On Post-Quantum Multi-Key Security of GCM

Akinori Hosoyamada[1,2]

[1] NTT Social Informatics Laboratories, Tokyo, Japan
[2] NTT Research Center for Theoretical Quantum Information, Atsugi, Japan
akinori.hosoyamada@ntt.com

Abstract. This paper studies the post-quantum multi-key security of Galois/Counter Mode (GCM) in the Quantum Ideal Cipher Model (QICM). GCM is one of the most widely deployed AEAD schemes. In practice, widely deployed cryptosystems are often instantiated under many independent keys, making the multi-key setting practically relevant. A trivial extension of a single-key security bound to the multi-key setting incurs a security loss proportional to the number of keys. In particular, in the post-quantum setting, the term corresponding to exhaustive key search becomes $up^2/2^k$, where u is the number of keys and k is the key length. Here, u is the number of keys, k is the key length, and p is the number of quantum queries to the underlying block cipher E and its inverse, which serves as a coarse measure of the amount of offline (quantum) computation performed by the adversary. For example, when $u = 2^{32}$, the trivial bound does not guarantee security for $p \geq 2^{48}$ when $k = 128$, and even for $k = 192$, it ceases to guarantee security for $p \geq 2^{80}$.

We show that, at the cost of some additional loss terms, the term $up^2/2^k$ can be replaced by a term of order $\sqrt{dp^2/2^k}$, where d denotes the maximum number of keys under which the same nonce appears in encryption queries. Thus, when d is much smaller than u (and the additional loss terms remain small), our bound improves upon the trivial multi-key bound. This can be viewed as a post-quantum counterpart of the classical result of Hoang et al. (CCS 2018). As in their work, we further show that, for randomized nonce-generation mechanisms that abstract patterns used in protocols such as TLS, the parameter d remains small even when u is large. Although our bounds are not tight and leave room for improvement, they yield a notable improvement over the trivial multi-key bound for several concrete parameter settings. To the best of our knowledge, this is the first non-trivial post-quantum multi-key security bound for an AEAD mode in the QICM. Our proof combines the reprogramming-and-resampling approach of Alagic et al. (EUROCRYPT 2022) with counting arguments used in the classical multi-user analysis of GCM by Hoang et al.

Keywords: GCM · post-quantum security · multi-key security · quantum ideal cipher model.

1 Introduction

Multi-Key Security in Symmetric Cryptography. The security of symmetric-key cryptographic schemes is often defined and analyzed in the single-key setting. In practice, however, the same symmetric-key primitive is deployed under many different keys, often across many users or sessions. It is therefore important to consider multi-key security, which is also commonly referred to as multi-user security [BBM00,Bih02]. The relevance of the multi-key setting is also recognized in standards and standards-related documents. NIST explicitly considers multi-key security in its recent specifications and evaluations of symmetric cryptography [TMC+25,Mou21], while RFC 9771 identifies multi-user security as a relevant security property of AEAD schemes, listing AES-GCM [MV04] as an example and protocols such as TLS, IPsec, and SRTP among its applications [Boz25].

Multi-key security is typically formulated by considering u independently keyed instances and declaring the adversary successful if it breaks any one of them. A trivial way to derive a multi-key security bound is to reduce multi-key security to single-key security. For example, if the single-key security advantage of a scheme is upper bounded by X, then its u-key security advantage is trivially upper bounded by uX.

Such a reduction, however, can incur an excessively large security loss. For example, (single-key) security bounds for block-cipher-based schemes typically contain a term of the form $\mathsf{Adv}^{\mathsf{prp}}_E + \cdots$, where $\mathsf{Adv}^{\mathsf{prp}}_E$ denotes the PRP advantage of the underlying block cipher E. If the key length of E is k-bit, the exhaustive key search with p evaluations of E or E^{-1} already achieves $\mathsf{Adv}^{\mathsf{prp}}_E \approx p/2^k$. The trivial multi-key reduction then yields a term of order $up/2^k$ in the security bound. For instance, when $k = 128$ and $u = 2^{32}$ keys are considered, the resulting bound no longer guarantees security once p reaches approximately 2^{96}.

In many practically deployed schemes, however, security does not necessarily degrade this rapidly as the number of keys increases. For GCM, for example, Bellare and Tackmann [BT16] and Hoang et al. [HTT18] showed that nonce randomization can substantially improve multi-user security in the ideal-cipher model. In particular, Hoang et al. showed that the problematic factor u in the term $up/2^k$ can essentially be avoided, yielding a term much closer to the single-key term $p/2^k$. Specifically, they first showed that if each nonce is used under at most d keys, then a term containing the factor $p/2^k$, corresponding to exhaustive key search, can depend on d rather than on the total number u of keys: namely, the relevant term becomes $dp/2^k$ instead of $up/2^k$. When the nonces contain sufficient randomness, d can remain small even when u is large, resulting in a term close to $p/2^k$. Separately, Luykx et al. [LMP17] showed that GCM as a mode incurs no multi-key security loss beyond that of the underlying block cipher.

These results are of significant practical importance. GCM is standardized by NIST [Dwo07] and ISO/IEC [ISO20], and is used in widely deployed protocols and standards including TLS [Res26], QUIC [TT21], IPsec [VM05], SSH [IS09].

Moreover, several protocols and standards employ nonce-generation mechanisms that provide the type of randomization relevant to the analyses in [BT16,HTT18].

Post-Quantum Security. The effect of the trivial multi-key reduction remains significant in the post-quantum setting. Due to Grover's algorithm [Gro96], the generic key-search term for the single-key PRP security of a block cipher deteriorates from $p/2^k$ in the classical setting to roughly $p^2/2^k$ in the quantum setting. If multi-key security is evaluated using the trivial reduction from single-key security, this immediately gives a term $up^2/2^k$. For example, taking $u = 2^{32}$ as an illustrative large-scale choice, the resulting bound no longer guarantees security for $p \geq 2^{48}$ when $k = 128$, and even when $k = 192$, it does not guarantee security for $p \geq 2^{80}$.

Some countries and organizations recommend secret-key lengths of $k \geq 192$ and hash output lengths of at least 384 bits to achieve post-quantum security [Age26,Dig26]. Viewed at the level of quantum query complexity as a coarse complexity measure, however, $p = 2^{80}$ lies below the roughly $2^{256/3} \approx 2^{85.3}$-query scale associated with BHT collision search for a 256-bit-output hash function [BHT98].

Setting $k = 256$ would improve security, but using a longer key can incur an efficiency cost. Thus, tighter bounds that justify a target security level with shorter keys can be practically useful.

The classical multi-key security results for GCM by Hoang et al. [HTT18] are proved in the classical ideal-cipher model and do not automatically extend to the post-quantum setting, where an adversary may make quantum queries to the underlying block cipher. Establishing a quantum analogue therefore requires an analysis in the Quantum Ideal Cipher Model (QICM) [HY18]. This leads us to the following question:

Can we avoid the term $up^2/2^k$ in a post-quantum multi-key security analysis of GCM, and obtain a security bound analogous to the classical bound of [HTT18]?

1.1 Our Contributions

In this paper, we answer this question affirmatively. More specifically, we prove a bound in which the quantum key-search term, which would trivially be $up^2/2^k$, is replaced by a term of order $\sqrt{dp^2/2^k}$, at the cost of additional loss terms. This can be viewed as a post-quantum counterpart of the classical result of Hoang et al. [HTT18]. To the best of our knowledge, our result gives the first non-trivial post-quantum multi-key security bound for an AEAD mode in the QICM.

We focus on the form of GCM in which nonces are mapped to block-cipher inputs by a key-independent injective encoding, rather than being processed through the keyed hash. This follows [HTT18] and includes, in particular, the standard 96-bit-IV instantiation of AES-GCM.

Our proof uses the reprogramming-and-resampling approach introduced by Alagic et al. [ABKM22] and has been applied to several symmetric-key constructions [ABK+24,Hos25,BBC+25,SI26]. Our result demonstrates that this approach can be applied even to the multi-key security analysis of an AEAD scheme, providing further evidence of the broad applicability of the reprogramming-and-resampling framework.

For privacy, in order to obtain the term $\sqrt{dp^2/2^k}$ instead of the trivial $up^2/2^k$ term, we combine the reprogramming-and-resampling framework with some of the counting arguments used in the classical H-coefficient proof [Pat09] of Hoang et al. [HTT18]. Using this combination, we prove the indistinguishability of the real encryption oracle from the ideal encryption oracle returning fresh random strings. For authenticity, we first use an argument analogous to the privacy proof to replace the real encryption oracle with the ideal one, and then show that the probability that any verification query is accepted with respect to the reprogrammed ideal cipher remains small.

Motivated by the discussion of nonce randomization in Hoang et al. [HTT18], we also analyze security when nonces are randomized. In particular, we consider nonce-generation mechanisms of the form "random salt + per-key distinct offsets (e.g., counter values)" and "random salt $\|$ nonce." The former (resp., latter) can be viewed as an abstraction of the nonce-generation patterns used in TLS 1.3 [Res26], QUIC [TT21], SRTP [MI15], and the successor-based instantiation of the RBG-based construction of NIST SP 800-38D [Dwo07] (resp., TLS 1.2 [SCM08] and IPsec ESP [VM05]).

Although our security bounds are somewhat involved, evaluating them under a concrete parameter setting illustrates a notable improvement over the trivial multi-key bound. In particular, using again the illustrative choice $u = 2^{32}$ considered above and setting $k = 192$, the p-dependent part of our bound reaches constant order at $p \approx 2^{92.8}$, in contrast to $p \approx 2^{80}$ for the trivial multi-key bound. (Here, this comparison refers to the quantum-query threshold at which the security upper bound becomes of constant order, rather than to a usage limit for any particular target advantage.)

Our security bounds are not tight and leave room for further improvement. We hope that this work motivates further study of post-quantum multi-key security for AEADs, as well as other schemes.

1.2 Related Works

Security of GCM (in the Single-Key Setting). Galois/Counter Mode (GCM) was introduced by McGrew and Viega together with its original security analysis [MV04]. Iwata et al. later identified a flaw in the proof, exhibited a distinguishing attack, and provided repaired privacy and authenticity bounds [IOM12]; Niwa et al. subsequently tightened these bounds substantially [NOMI15]. The tightness of GCM's authenticity bounds and corresponding forgery attacks was further investigated by Luykx and Preneel [LP18] and Nandi [Nan18]. GCM has also been studied under stronger robustness notions

in, e.g., nonce-misuse and nonce-misuse resilience [Jou06,ADL17], authenticity under RUP [IIM25], and committing security [DGRW18,CR22], to name a few.

Variants of GCM. Various variants of GCM have been proposed to improve its misuse resistance, concrete security, robustness, or nonce and key handling. GCM-SIV was introduced by Gueron and Lindell as an efficient nonce-misuse-resistant AEAD based on GCM-like components [GL15,GLL17]; its security was further studied and strengthened by Iwata and Minematsu [IM16] and Iwata and Seurin [IS17], while Bose et al. established improved multi-user bounds for AES-GCM-SIV [BHT18]. Chung et al. proposed eGCM and eGCM-SIV, which target almost full security while supporting longer nonces or nonce-misuse resistance [CHK+25]. Bhattacharya and Nandi proposed mGCM, a simple variant of GCM based on a variable-output-length pseudorandom function that provides stronger concrete security [BN18]. Bellare and Hoang proposed modifications of GCM and AES-GCM-SIV that additionally provide committing security [BH22]. GCM-SST, specified by Campagna et al. [CMP24], is designed to provide stronger authentication security, in particular for truncated tags; its generic and multi-user security have been analyzed by Inoue et al. [IJMM25] and Naito et al. [NSS25], respectively.

Randomized-nonce variants of GCM, including RGCM and XGCM, and their multi-user security were studied by Bellare and Tackmann [BT16] and Hoang et al. [HTT18]. Another approach to improving the security bounds of GCM is nonce-based key derivation, as studied by Gueron and Lindell [GL17]. More recently, Gueron and Ristenpart proposed the derived-nonce, derived-key (DNDK) framework and its concrete instantiation DNDK-GCM, aiming to improve the secure scalability of GCM [GR25]. Naito et al. proposed nonce-based and tag-based key derivation (NTKD) and applied it to GCM and CCM to obtain improved multi-user security for large amounts of data [NSS26]. Bhaumik et al. proposed ♯Pencil, domain-extended pseudorandom functions that can be used to strengthen GCM through key derivation while accommodating larger nonce domains [BDD26].

Related directions have also been considered in the ongoing revision of NIST SP 800-38D. NIST has considered nonce-key-derivation proposals [Nat25], and has subsequently proposed extending GCM to 256-bit-block ciphers under the name wGCM [Nat26].

Our work instead studies the post-quantum multi-key security of standard GCM itself, without modifying the underlying mode, while additionally analyzing randomized nonce-generation mechanisms.

Post-Quantum Security of Symmetric-Key Modes and AEAD. Post-quantum security of symmetric-key modes has been studied under several oracle models. Anand et al. analyzed standard block-cipher modes against quantum adversaries [ATTU16], while Soukharev et al. studied post-quantum authenticated encryption more generally [SJS16]; post-quantum security has also been established for keyed sponge-based constructions including Ascon [Hos25]. Most

closely related to our setting, Alagic et al. obtain post-quantum security bounds for block-cipher based modes including GCM in the single-key QICM [ABMS26], whereas Shiba and Iwata study post-quantum multi-key security of tweakable Even–Mansour [EM97] and FX [KR96] in the Q1MK model [SI26]. Our work lies at the intersection of these directions, treating GCM as an AEAD in the post-quantum multi-key setting.

A separate line of work allows quantum superposition access to the construction itself. Kaplan et al. showed that modes including GMAC and GCM become insecure in such a setting [KLLN16], while QCB provides a positive construction designed for quantum superposition queries [BBC+21]. Lang and Lucks further studied GCM and other classical AE schemes under such attacks [LL23], and Fischlin and Schmalz recently proposed quantum-resistant variants of GCM and GCM-SIV achieving security in a fully quantum model [FS26]. In contrast, our encryption and verification interfaces are classical, while quantum access is given only to the underlying ideal cipher.

2 Preliminaries

For a bit string x, by $\mathsf{msb}_i(x)$ (resp., $\mathsf{lsb}_i(x)$) we denote the most significant i bits (resp., least significant i bits) of x. We often identify n-bit strings x with integers such that $0 \leq x < 2^n$. When x and y are n-bit strings or integers such that $0 \leq x, y < 2^n$, the value $x + y$ is understood as their sum modulo 2^n, unless otherwise noted. For a finite set $\mathcal{X}$, let $x \xleftarrow{\$} \mathcal{X}$ denote that an element x is sampled from $\mathcal{X}$ uniformly at random. For any distribution D over a finite set $\mathcal{X}$, we denote its min-entropy by $H_\infty(D)\,(= -\log_2 \max_{x\in\mathcal{X}} D(x))$.

An n-bit block cipher with k-bit keys is a function $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ such that $E(K,\cdot)$ is a permutation for all $K \in \{0,1\}^k$. $\mathsf{BC}(n,k)$ denotes the set of all such E. By E^{-1} we denote the inverse cipher satisfying $E^{-1}(K, E(K,x)) = x$ for all K and x. We often write $E_K(x)$ and $E_K^{-1}(y)$ instead of $E(K,x)$ and $E^{-1}(K,y)$.

Quantum Computation. We assume that the readers are familiar with the basics of quantum computation. The quantum oracle of a function $f : \mathcal{X} \to \mathcal{Y}$ is given as the unitary operator $U_f : |x,y\rangle \to |x, y : f(x)\rangle$ (we assume that $\mathcal{Y}$ is an Abelian group with a group operation $+$). When we talk about a security notion in the Quantum Ideal Cipher Model (QICM), we assume that an $E \in \mathsf{BC}(n,k)$ is chosen uniformly at random at the beginning of a security game, and an adversary is given the quantum oracle access to E and E^{-1}. We often write like $\mathcal{A}^{|E^{\pm}\rangle}$ to indicate that this oracle access is given to an adversary $\mathcal{A}$.

2.1 Almost XOR Universal Hash Functions

A keyed hash function with a key space $\mathcal{K}$, an input space $\mathcal{X}$, and an output $\mathcal{Y}$, is a function $H : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We often write $H_K(X)$ instead of $H(K,X)$.

Following [HTT18], we define almost XOR universal hash functions as follows: Let n be a positive integer, $\mathcal{X} := \{0,1\}^* \times \{0,1\}^*$, $\mathcal{Y} := \{0,1\}^n$. For a value $c \geq 1$, H is called a c-almost XOR universal hash (c-AXU) if it satisfies

$$\Pr_K [H_K(X_1, X_2) \oplus H_K(X_1', X_2') = Y] \leq c \cdot \frac{\max\{|X_1|_n + |X_2|_n, |X_1'|_n + |X_2'|_n\}}{2^n}$$

for any distinct $(X_1, X_2), (X_1', X_2') \in \{0,1\}^* \times \{0,1\}^*$, and any $Y \in \{0,1\}^n$. Here, the probability is taken over a uniform random choice of $K \in \mathcal{K}$, and $|X|_n := \max\{1, \lceil |X|/n \rceil\}$.

2.2 Authenticated Encryption with Associated Data

Syntax and Notations. An Authenticated Encryption with Associated Data (AEAD) scheme Π is a pair of functions $\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T}$ and $\mathcal{D} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\}$. Here, $\mathcal{K}$ is a key space, $\mathcal{N}$ is a nonce space, $\mathcal{AD}$ is a space of associated data, $\mathcal{M}$ is a message space, $\mathcal{C}$ is a ciphertext space, and $\mathcal{T}$ is a tag space, all of which are sets of some bit strings. We require that $\mathcal{D}(K, N, AD, \mathcal{E}(K, N, AD, M)) = M$ for any $(K, N, AD, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{M}$, and assume that $|C| = |M|$ when $\mathcal{E}(K, N, AD, M) = (C, T)$ for some K, N, AD, T. We often write $\mathcal{E}_K(N, AD, M)$ (resp., $\mathcal{D}_K(N, AD, C, T)$) instead of $\mathcal{E}(K, N, AD, M)$ (resp., $\mathcal{D}(K, N, AD, C, T)$).

In addition, the verification function $\mathcal{V} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{C} \times \mathcal{T} \to \{\top, \perp\}$ is defined by $\mathcal{V}(K, N, AD, C, T) = \top$ if $\mathcal{D}(K, N, AD, C, T) \neq \perp$, and $\mathcal{V}(K, N, AD, C, T) = \perp$ if $\mathcal{D}(K, N, AD, C, T) = \perp$.

When multiple keys $K_1, \ldots, K_u \in \mathcal{K}$ are considered, we define Enc and Vrf by $\mathsf{Enc}(\mathsf{id}, N, AD, M) := \mathcal{E}_{K_{\mathsf{id}}}(N, AD, M)$ and $\mathsf{Vrf}(\mathsf{id}, N, AD, C, T) := \mathcal{V}_{K_{\mathsf{id}}}(N, AD, C, T)$, respectively ($\mathsf{id} = 1, \ldots, u$).

This paper considers AEADs that are built on top of a block cipher E. We write that Π^E, $\mathcal{E}^E$, Enc^E, and so on, to indicate that the schemes depend on E.

Security (in the QICM). We study privacy and authenticity separately. Let $\Pi^E = (\mathcal{E}^E, \mathcal{D}^E)$ be an AEAD scheme built on top of a block cipher E. We consider security in the multi-key setting (also commonly referred to as the multi-user setting)[3], where u keys $K_1, \ldots, K_u \in \mathcal{K}$ are chosen uniformly at random for some u (in the real world), and the oracles Enc^E and Vrf^E perform the encryption and verification with respect to these u keys. We discuss security in the Quantum Ideal Cipher Model (QICM), where an adversary is given the quantum oracle of an ideal cipher E and its inverse $E^{\pm}$. We refer to the oracles of E/E^{-1} the primitive oracles, and the queries to them as primitive queries.

[3] We use the term "multi-key" instead of "multi-user" because the indices in our model refer to independently generated keys rather than physical users; in particular, a single user may use multiple keys.

Privacy. The multi-key privacy (mk-priv) advantage of an adversary $\mathcal{A}$ in the QICM is defined by

$$\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\Pi}(\mathcal{A}) := \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{Enc}^E,|E^{\pm}\rangle}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\$,|E^{\pm}\rangle}\right]\right|.$$

Here, \$ is the oracle that returns a random string of length $|M|+\tau$ for any input (i, N, AD, M), and the keys $K_1, \ldots, K_u$ for the real encryption oracle are chosen independently and uniformly at random. The oracles of Enc^E and \$ are classical, while the access to the ideal cipher E and its inverse is quantum. We refer to Enc^E and \$ as the real and ideal encryption oracles, respectively, and queries to them as encryption queries. We focus on the nonce-respecting setting: For each $(i, N) \in \{1, \ldots, u\} \times \mathcal{N}$, once $\mathcal{A}$ queries (i, N, AD, M) to the encryption oracle for some (AD, M), it is prohibited to query (i, N, AD', M') afterwards for any (AD', M').

Authenticity. Following prior analyses of GCM [IOM12], we consider an online authenticity notion in which the adversary may make multiple forgery attempts. We use an accept/reject verification oracle, as in the multi-user AE formulations of [BT16,HTT18], rather than a decryption oracle.

The multi-key authenticity (mk-auth) advantage of an adversary $\mathcal{A}$ in QICM is defined by

$$\mathsf{Adv}^{\mathsf{mk\text{-}auth}}_{\Pi}(\mathcal{A}) := \Pr\left[\mathcal{A}^{\mathsf{Enc}^E,\mathsf{Vrf}^E,|E^{\pm}\rangle} \text{ forges }\right],$$

where "$\mathcal{A}^{\mathsf{Enc}^E,\mathsf{Vrf}^E,|E^{\pm}\rangle}$ forges " means that the oracle Vrf^E returns $\top$ while $\mathcal{A}$ is running, and the keys $K_1, \ldots, K_u$ for the real encryption oracle are chosen independently and uniformly at random. The oracles of Enc^E and Vrf^E are classical, while the oracle of E / E^{-1} are quantum. To avoid trivial wins, we prohibit $\mathcal{A}$ from forwarding the response from the encryption oracle to the verification oracle. We refer to the oracle Vrf^E as the verification oracle, and queries to it as verification queries.

As in the definition of privacy, we require that $\mathcal{A}$ is nonce-respecting for the encryption oracle, i.e., it is prohibited to query the same pair (i, N) to the encryption oracle. (This restriction is not applied to verification queries.)

2.3 Reprogramming and Resampling Lemmas

Our proof is based on the resampling-and-reprogramming approach that was introduced by Alagic et al. [ABKM22] and has been applied in several follow-up works [ABK+24,Hos25,BBC+25,SI26]. Their core technical lemmas are the arbitrary reprogramming lemma and the resampling lemma, which we recall below (for the resampling lemma, we use a variant shown in [ABMS26]).

Lemma 1 (Arbitrary reprogramming lemma [ABKM22]). *Let $\mathcal{A}'$ be a quantum algorithm that runs in the following three-stage game.*

1. *After performing some computation, $\mathcal{A}'$ outputs the descriptions of a function $F_0 : \{0,1\}^m \to \{0,1\}^n$ and a randomized algorithm $\mathcal{B}$. Here, $\mathcal{B}$ is an algorithm that outputs another function $F_1 : \{0,1\}^m \to \{0,1\}^n$.*
2. *Run $\mathcal{B}$ to sample F_1, sample a bit b uniformly at random, and resume running $\mathcal{A}'$ relative to the quantum oracle of F_b.*
3. *Provide $\mathcal{A}'$ with the randomness used to invoke $\mathcal{B}$, and continue running $\mathcal{A}'$ without an oracle. Finally, $\mathcal{A}'$ outputs a bit b'.*

Suppose that the expected value of the number of the quantum queries by $\mathcal{A}'$ is at most Q when $b = 0$. Then,

$$|\Pr[1 \leftarrow \mathcal{A}'|b = 0] - \Pr[1 \leftarrow \mathcal{A}'|b = 1]| \leq 2Q\sqrt{\varepsilon}$$

holds, where $\varepsilon := \max_{x\in\{0,1\}^m} \Pr[F_1(x) \neq F_0(x)]$.

For $K_* \in \{0,1\}^k$ and $s_0, s_1 \in \{0,1\}^n$, define a function $\mathsf{swap}^{K_*}_{s_0,s_1} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k \times \{0,1\}^n$ by

$$\mathsf{swap}^{K_*}_{s_0,s_1}(K,x) = \begin{cases} (K,x) & \text{if } K \neq K_* \text{ or } K = K_* \wedge x \notin \{s_0,s_1\}, \\ (K, s_{b\oplus 1}) & \text{if } K = K_* \text{ and } x = s_b. \end{cases} \tag{1}$$

Lemma 2 (Resampling lemma for ideal ciphers [ABMS26]). *Let E be an ideal cipher chosen from $\mathsf{BC}(n,k)$ uniformly at random, and $\mathcal{A}'$ be a quantum adversary that runs in the following two-stage game.*

1. *$\mathcal{A}'$ runs relative to the quantum oracles of E and E^{-1}. After performing some computation and making some queries, $\mathcal{A}'$ outputs a distribution D over $\{0,1\}^k \times \{0,1\}^n \times \{0,1\}^n$.*
2. *A tuple (K_*, s_0, s_1) is sampled according to the distribution D, and a bit b is sampled uniformly at random. Set $E^{(0)} := E$ and $E^{(1)} := E \circ \mathsf{swap}^{K_*}_{s_0,s_1}$. Give (K_*, s_0, s_1) to $\mathcal{A}'$, and continue running $\mathcal{A}'$ relative to the quantum oracles of $E^{(b)}$ and its inverse. Finally, $\mathcal{A}'$ outputs a bit b'.*

If $\mathcal{A}'$ makes at most p quantum queries at the first stage, then

$$|\Pr[1 \leftarrow \mathcal{A}'|b = 0] - \Pr[1 \leftarrow \mathcal{A}'|b = 1]| \leq 4\sqrt{p \cdot 2^{n - H_\infty(D)}}.$$

2.4 Some Balls-into-Bins Lemmas

Like the classical work of Hoang et al. [HTT18], we also use the following two balls-into-bins lemmas.

Lemma 3 ([BHT18]). *Let n, ℓ, a, and q be integers such that $n \geq 128$, $\ell \geq 2$, $a \geq 1$, and $q \leq a \cdot 2^n$. Let $X_1, \ldots, X_q$ be random variables taking values in $\{0,1\}^n$, which may not be independent from each other but satisfy*

$$\Pr[X_i = Y|E] \leq 2/2^n$$

for any $Y \in \{0,1\}^n$ and all $i = 1, \ldots, q$, and for any event E on $X_1, \ldots, X_{i-1}$. Then,

$$\Pr\left[\exists Y \in \{0,1\}^n : \#\{i : X_i = Y\} \geq \lceil a\ell n/2 \rceil\right] \leq 2^{-1.5\ell n}.$$

Lemma 4 ([HTT18]). *Let r, u be positive integers, ε be a number such that $0 < \varepsilon < 1$, and q be a positive integer such that $q \leq 2^{(1-\varepsilon)r}$. Let $I_1, \ldots, I_u$ be a partition of $I := \{1, \ldots, q\}$ (i.e., $I_a \cap I_b = \varnothing$ and $I_1 \cup \cdots \cup I_u = \{1, \ldots, q\}$). Let X_j^i be random variables, indexed by $i \in \{1, \ldots, u\}$ and $j \in I_i$, that take values in $\{0,1\}^r$. Assume that the two variables X_j^i and $X_{j'}^{i'}$ are independent for $i \neq i'$, the marginal distribution of each X_j^i is uniform over $\{0,1\}^r$, and $X_j^i \neq X_{j'}^i$ for $j \neq j'$. In addition, let $d := \lceil 1.5/\varepsilon \rceil - 1$. Then,*

$$\Pr\left[\exists Y \in \{0,1\}^r : \#\left\{(i,j) : X_j^i = Y\right\} > d\right] \leq 2^{-r/2}.$$

3 Description of CAU

Here we describe an AEAD scheme called CAU [BT16,HTT18], which abstracts the form of GCM [MV04,Dwo07] in which nonces are embedded into block-cipher inputs without being hashed. For 128-bit block ciphers and 96-bit nonces, this coincides with the standard processing of 96-bit nonces in AES-GCM. We do not consider the alternative GCM processing for other nonce lengths, in which the nonce is first processed using the keyed hash.

CAU is built on top of a block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ and a keyed hash function $H : \{0,1\}^n \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^n$. The key space of CAU is $\{0,1\}^k$. The message and associated data spaces are both $\{0,1\}^*$. The nonce and tag spaces are $\{0,1\}^r$ and $\{0,1\}^\tau$ for some $r < n$ and $\tau \leq n$, respectively. The encryption and decryption algorithms of CAU are defined as follows.

Encryption algorithm of CAU.

1. Suppose that a key $K \in \{0,1\}^k$, nonce $N \in \{0,1\}^r$, associated data $AD \in \{0,1\}^*$, and a message $M \in \{0,1\}^*$ are given as input. Compute $Y := \mathsf{pad}(N)$. If M is not the empty string ($|M| \neq 0$), parse M into blocks as $M = M[1]||\cdots||M[\ell]$, where $|M[i]| = n$ for $i < \ell$ and $1 \leq |M[\ell]| \leq n$. Here, pad is the function from $\{0,1\}^r$ to $\{0,1\}^n$ defined by $\mathsf{pad}(N) = N||0^{n-r-1}||1$.
2. If M is the empty string ($|M| = 0$), then set C as the empty string. If M is not the empty string, then compute the ciphertext C by using the counter mode with the initial counter $(Y+1)$. That is, compute $V[i] := E_K(Y+i)$, $C[i] := M[i] \oplus V[i]$ for $i = 1, \ldots, \ell - 1$, and $C[\ell] := M[\ell] \oplus \mathsf{msb}_{|M[\ell]|}(V[\ell])$. Then, set $C := C[1]||\cdots||C[\ell]$.
3. Compute the tag T by hashing the ciphertext with the Wegman-Carter MAC based on H and the hash key $E_K(0)$. That is, set $L := E_K(0)$, and compute $T := \mathsf{msb}_\tau(H_L(AD, C) \oplus E_K(Y))$.
4. Return (C, T).

Decryption algorithm of CAU.

1. Suppose that a key $K \in \{0,1\}^k$, nonce $N \in \{0,1\}^r$, associated data $AD \in \{0,1\}^*$, a ciphertext $C \in \{0,1\}^*$ and a tag $T \in \{0,1\}^\tau$ are given as input. Compute $Y := \mathsf{pad}(N)$. If C is not the empty string ($|C| \neq 0$), parse C into blocks as $C = C[1]||\cdots||C[\ell]$, where $|C[i]| = n$ for $i < \ell$ and $1 \leq |C[\ell]| \leq n$.

2. Verify if the input is authentic. That is, compute $L := E_K(0)$ and $T' := \mathsf{msb}_\tau\left(H_L(AD, C) \oplus E_K(Y)\right)$, and return $\perp$ if $T \neq T'$. If $T = T'$, then proceed to the next step.
3. If C is the empty string ($|C| = 0$), then set M as the empty string. If C is not the empty string, compute the message M by using the counter mode with the initial counter $(Y + 1)$. That is, compute $V[i] := E_K(Y + i)$ and $M[i] := C[i] \oplus V[i]$ for $i = 1, \ldots, \ell - 1$, and $M[\ell] := C[\ell] \oplus \mathsf{msb}_{|C[\ell]|}(V[\ell])$. Then, set $M := M[1]||\cdots||M[\ell]$.
4. Return M.

We count the length of messages by n-bit blocks. For example, we regard that a non-empty message $M = M[1]||\cdots||M[\ell]$ consists of $\ell := \lceil |M|/n \rceil$ blocks, where $|M[i]| = n$ for $i < \ell$ and $|M[\ell]| \leq n$. We refer to ℓ as the block length of M, and $M[i]$ as the ith block of M. (The block length of the empty message is defined to be zero.) We also use the same terminology for other variable length variables appearing in the encryption and decryption algorithms, e.g., associated data, ciphertexts, etc.

In what follows, we always assume that $k \geq n \geq 128$, and that the block length of the messages M is at most $2^{n-r} - 2$. (Due to the latter assumption, when considering nonce-respecting security, the value $E_K(x)$ is evaluated at most once in the real encryption oracle throughout the security games for each key index and each $x \neq 0$.)

4 Post-Quantum Privacy of CAU in the QICM

Regarding the post-quantum privacy of CAU, the following theorem holds.

Theorem 1. *Suppose $u \leq 2^k$, and H is a c-AXU. Suppose that $\mathcal{A}$ makes at most q_e classical encryption queries such that the length of the messages is at most σ_e blocks in total, and that $\mathcal{A}$ queries every single nonce to the encryption oracle for at most d keys. Additionally, suppose that $\mathcal{A}$ makes at most Q_e classical queries for each key, the length of messages for each key is at most B_e blocks in total, and that the block length of (AD, C) (input to H) is at most $\ell^H_{\max}$ for every encryption query. Moreover, assume that the number of quantum queries by $\mathcal{A}$ to the ideal cipher E and the inverse E^{-1} is at most p. Then, we have $\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}}$, where*

$$\begin{aligned}\delta_{\mathsf{priv}} :=\ & 4(\sigma_e + q_e)\sqrt{\frac{p}{2^k}} + 2p\sqrt{\frac{4d}{2^k}} \\ & + \frac{16(\sigma_e + q_e)\ell^H_{\max}}{2^{(\ell^H_{\max}-2)n/2}} + \frac{2(\sigma_e + q_e)}{2^{2n}} + \frac{4(\sigma_e + q_e + 1)^{d+2}}{(d+1)!\,2^{(d-1)n}} \\ & + \frac{2\left(n\ell^H_{\max} + 8\right)(\sigma_e + q_e)Q_e + 2\,(n+8)\,(\sigma_e + q_e)B_e + 4(\sigma_e + q_e + 1)^2}{2^{n+1}} \\ & + \frac{2(\sigma_e + q_e)(Q_e + B_e + 2d)}{2^k}. \end{aligned} \tag{2}$$

Proof Overview. Our proof follows the reprogramming-and-resampling paradigm introduced by Alagic et al. [ABKM22] and have been applied in several previous works [ABK+24,Hos25,BBC+25,SI26]. Roughly speaking, we replace the response blocks of the real encryption oracle by fresh random blocks one by one. After each replacement, we reprogram the ideal cipher E so that its evaluations under the secret keys $K_1, \ldots, K_u$ remain consistent with the classical encryption transcript generated so far. Formally, this is implemented by a sequence of hybrids indexed by the number of response blocks that have already been replaced.

An important point is that we never reprogram the values $E_{K_i}(0)$, which serve as the hash keys. Indeed, the input 0 is shared by all u keys, and hence the assumption that a single nonce is used under at most d keys gives no useful control at this point. Naively reprogramming $E_{K_i}(0)$ for all u keys would therefore reintroduce a quantum security loss depending explicitly on u, with essentially the same multi-key query threshold as the trivial bound $up^2/2^k$. Instead, we leave $E_{K_i}(0)$ untouched and analyze the resulting dependence between the hash function and the ideal cipher directly.

Like the previous works relying on the reprogramming-and-resampling approach, there are two main sources of quantum-specific security loss in the proof. First, when a fresh random response block is introduced, the resampling lemma (Lemma 2) is used to modify one corresponding value of the ideal cipher. Applying this argument block by block gives the term of order $(\sigma_e+q_e)\sqrt{p/2^k}$. Second, after a response block has been fixed, we use the arbitrary reprogramming lemma (Lemma 1) to shift the timing of the reprogramming from immediately after that block to just before the next response block is generated. The most important thing is that, except for unlikely collision events, the number of secret keys that can cause the same relevant reprogramming point is controlled by the parameter d, rather than by the total number u of keys. This yields the term of order $p\sqrt{d/2^k}$.

The technically most involved part is establishing this latter property while ensuring that all reprogramming operations are mutually consistent. We define bad events corresponding to collisions among the input/output points involved in the reprogramming operations and bound their probabilities. For this purpose, we combine the quantum reprogramming argument with maximum-load and balls-into-bins techniques appearing in the classical multi-key analysis of CAU [HTT18]. The proof framework, however, is substantially different: the analysis of [HTT18] is based on the H-coefficient technique, whereas ours proceeds by gradually resampling and reprogramming the ideal cipher. Thus, incorporating their classical counting arguments into our proof requires controlling how the corresponding collision events interact with the dynamically reprogrammed points. In particular, these arguments control the number of keys K satisfying transcript-induced relations of the forms $E_K(x) = y$ and $E_K(x) \oplus H_{E_K(0)}(X) = y$. Together with a bound on the multiplicity of values appearing in the random encryption transcript, this allows us to replace a

potential dependence on the total number of keys u by the nonce-overlap parameter d.

Modification to the Encryption Oracles. Since the classical verification oracle is not given to the adversary when considering privacy, truncating tag bits only increases the difficulty to distinguish the real and ideal encryption oracles. Hence, in what follows, we prove the claim when $\tau = n$.

Moreover, we slightly modify the encryption oracles as follows (suppose a tuple (i, N, AD, M) is queried to the encryption oracle):

1. In the real encryption oracle, each message M is first padded with zeroes at the end so the length will be a multiple of n. The resulting message is denoted by $\bar{M}$ (i.e., $\bar{M} = M||0\cdots0$ and $|\bar{M}|/n = \lceil |M|/n \rceil$). Then, the ciphertext C is computed for this padded $\bar{M}$, and returned to the adversary. The computation of the tag is the same as before: some least significant bits of C are truncated so the length will be equal to $|M|$ (the resulting string is denoted by $\tilde{C}$), the hash value $H_L(AD, \tilde{C})$ is computed, and then $T := H_L(AD, \tilde{C}) \oplus E_K(Y)$ is returned as the tag.
2. In the ideal encryption oracle, uniform random n-bit blocks $C[1], C[2], \ldots, C[\ell], T$ are sampled in sequential order, where ℓ is the block length of the queried message. Then, $C := C[1]||\cdots||C[\ell]$ and T are returned to the adversary.

This change only increases the distinguishing advantage of the adversary because the information given to the adversary is increased. Hence, in what follows, we prove the indistinguishability of these two modified oracles. In particular, note that the length of C is always a multiple of n, and the truncated "ciphertext" given to the hash function H is now denoted by $\tilde{C}$. We will also use the symbol $\bar{M}$ to denote the padded message defined as above.

Remark 1. The above modifications are somewhat complicated, but they are not essential, and the oracles' behavior is unchanged under the condition that $\mathcal{A}$ queries only messages the length of which is a multiple of n. We recommend that readers assume this condition first when reading the proof for the first time.

Notations and Terminology. For each response (C, T) from an encryption oracle, we refer to $C[i]$ as the ith block of the response for $i \leq \ell$, and T as the $(\ell+1)$th block (or, just the last block). We sometimes count the number of blocks in the response from the beginning of the game. For example, if $\mathcal{A}$ makes two classical queries to the encryption oracle and the responses are (C_1, T_1) and (C_2, T_2), then the first block in (C_2, T_2) is the (ℓ_1+2)nd block (with $\ell_1 := |C_1|/n$).

Transcript. Let $\mathsf{T}_z^{\mathsf{enc}}$ be the transcript of the classical encryption queries and the responses up to the zth block in the responses (in the ideal world), counted from the beginning of the game. That is,

$$\mathsf{T}_z^{\mathsf{enc}} = (((\mathsf{id}_1, N_1, AD_1, M_1), (C_1, T_1)), \ldots, ((\mathsf{id}_i, N_i, AD_i, M_i), (C_i, T_i))) \quad (3)$$

for some i, where $(\mathsf{id}_j, N_j, AD_j, M_j)$ denotes the jth encryption query (resp., (C_j, T_j) denotes the response) for each j. Here, the block length of the responses is z in total, and (C_i, T_i) may be incomplete; (C_i, T_i) contains only from its first block to the $(z - (\ell_1 + \cdots + \ell_{i-1} + i - 1))$th block (here, ℓ_j is the block length of M_j and C_j).

For each j and $w \leq \ell_j$, we also define $V_j[w] := \bar{M}_j[w] \oplus C_j[w]$ as long as $C_j[w]$ is defined (note that this $V_j[w]$ matches the corresponding variable in the description of CAU on page 10). Similarly, let $Y_j := \mathsf{pad}(N_j)$, and L_j be the hash key in the tag computation (i.e., $L_j := E_{K_{\mathsf{id}_j}}(0)$).

For each j, except for the hash keys L_j, the block cipher E is evaluated $(\ell_j + 1)$ times in the jth encryption query; $E_{K_{\mathsf{id}_j}}(Y_j + 1), \ldots, E_{K_{\mathsf{id}_j}}(Y_j + \ell_j)$ are evaluated in the counter mode to compute the ciphertext C_j, and then $E_{K_{\mathsf{id}_j}}(Y_j)$ is evaluated when computing the tag T_j. The total number of evaluations of E until the ith encryption query except for the hash keys is $(\sigma_i + i)$, where $\sigma_i := \ell_1 + \cdots + \ell_i$.

Reprogramming of E. In the proof, we will reprogram E so its outputs will be compatible with (a part of) the transcript. Let $\mathsf{T}_z^{\mathsf{enc}}$ be the transcript as in Eq. (3), i be also as the index in Eq. (3), and $\mathbf{K} := (K_1, \ldots, K_u)$ be keys for E.

For each $j < i$ and $w \leq \ell_j + 1$, define $\mathsf{ReProg}_j^{\leq w} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k \times \{0,1\}^n$ by

$$\mathsf{ReProg}_j^{\leq w} := \begin{cases} \text{identity map} & \text{for } w = 0, \\ \mathsf{swap}^{K_{\mathsf{id}_j}}_{Y_j+w, E^{-1}_{K_{\mathsf{id}_j}}(V_j[w])} \circ \cdots \circ \mathsf{swap}^{K_{\mathsf{id}_j}}_{Y_j+1, E^{-1}_{K_{\mathsf{id}_j}}(V_j[1])} & \text{for } 1 \leq w \leq \ell_j \\ \mathsf{swap}^{K_{\mathsf{id}_j}}_{Y_j, E^{-1}_{K_{\mathsf{id}_j}}(T_j \oplus H_{L_j}(AD_j, \tilde{C}_j))} \circ \mathsf{ReProg}_j^{\leq \ell_j} & \text{for } w = \ell_j + 1. \end{cases} \tag{4}$$

For $j = i$, we also define $\mathsf{ReProg}_j^{\leq w}$ analogously, as long as it can be defined, and let $\mathsf{FinBlock}$ be the number of the blocks contained in (C_i, T_i) (that is, $\mathsf{FinBlock} := z - (\ell_1 + \cdots \ell_{i-1} + i - 1)$.

Let $E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]$ be the block cipher defined by

$$E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!] := E \circ \mathsf{ReProg}_i^{\leq \mathsf{FinBlock}} \circ \mathsf{ReProg}_{i-1}^{\leq (\ell_{i-1}+1)} \circ \cdots \circ \mathsf{ReProg}_1^{\leq (\ell_1+1)} \tag{5}$$

for $z \geq 1$, and $E[\![\mathsf{T}_0^{\mathsf{enc}}, \mathbf{K}]\!] := E$.

Intuitively, $\mathsf{ReProg}_j^{\leq w}$ reprograms[4] the values of E so that they are compatible with the response to the jth encryption query up to the wth block (except for the computation of the hash key $L_j = E_{K_{\mathsf{id}_j}}(0)$). In $E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]$, the output values of E are reprogrammed so they are compatible with the responses to the classical encryption queries, up to the zth block counted from the beginning of the game (except for the evaluations to compute hash keys).

[4] ChatGPT was used to suggest the choice of the value to be reprogrammed at this point. The author subsequently worked out and verified the resulting argument.

Hybrids. To prove the theorem, we introduce hybrids $\mathbf{H}_1^z, \ldots, \mathbf{H}_4^z$ as follows for $z = 0, \ldots, \sigma_e + q_e$. (We define each hybrid so that $\mathcal{A}$ runs relative to the oracles in the ideal world and pause $\mathcal{A}$ at some point, and then modify some oracles. If $\mathcal{A}$ finishes running and returns the final output before reaching that point of the hybrid, it is understood that $\mathcal{A}$ runs just as in the ideal world.)

$\mathbf{H}_1^z$. In this hybrid, the adversary $\mathcal{A}$ runs relative to the ideal encryption oracle \$ and the quantum ideal cipher oracle $|E^{\pm}\rangle$, and we pause $\mathcal{A}$ just before the $(z+1)$th block (counted from the beginning) in the response to the classical queries is sampled. Then, we sample u keys $\mathbf{K} = (K_1, \ldots, K_u)$ from $\{0,1\}^k$ uniformly at random, switch the computation of the encryption responses from \$ to $\mathsf{Enc}^{E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]}$ (resp., replace the quantum oracles of E and its inverse with those of $E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]$ and its inverse), and continue running $\mathcal{A}$ until it returns the final output.

$\mathbf{H}_2^z$. This hybrid is identical to $\mathbf{H}_1^z$ until $\mathcal{A}$ is paused. This is just before the $(z+1)$th block is sampled. Suppose that this block is the wth block in the response to the current encryption query, denoted by (id, N, AD, M). When $\mathcal{A}$ is paused, sample u keys $\mathbf{K} = (K_1, \ldots, K_u)$ from $\{0,1\}^k$ uniformly at random, sample $s_1 \in \{0,1\}^n$ uniformly at random, and define a block cipher E' by

$$E' := \begin{cases} E \circ \mathsf{swap}_{Y+w, s_1}^{K_{\mathsf{id}}} & \text{if } w \le \ell, \\ E \circ \mathsf{swap}_{Y, s_1}^{K_{\mathsf{id}}} & \text{if } w = \ell + 1, \end{cases}$$

where $Y := \mathsf{pad}(N)$ and ℓ be the block length of M. Replace the oracle \$ (resp., the quantum oracles of E and its inverse) with $\mathsf{Enc}^{E'[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]}$ (resp., the quantum oracles of $E'[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]$ and its inverse), and continue running $\mathcal{A}$ until it returns the final output.

$\mathbf{H}_3^z$. This hybrid is a slight modification of $\mathbf{H}_2^z$: In this hybrid, the $(z+1)$th block in the responses is sampled uniformly at random (which is the ciphertext block $C[w]$ if $w \le \ell$ and the tag block T if $w = \ell + 1$), and s_1 is defined depending on this sampled block by $s_1 := E_{K_{\mathsf{id}}}^{-1}(C[w] \oplus \bar{M}[w])$ if $w \le \ell$ and $s_1 := E_{K_{\mathsf{id}}}^{-1}(T \oplus H_L(AD, \tilde{C}))$ for $w = \ell + 1$ ($L := E_{K_{\mathsf{id}}}(0)$). Except for these modifications, $\mathbf{H}_3^z$ is identical to $\mathbf{H}_2^z$. (As mentioned before, $\tilde{C}$ denotes the truncated version of C such that $|\tilde{C}| = |M|$.)

$\mathbf{H}_4^z$. In this hybrid, the adversary $\mathcal{A}$ runs relative to the ideal encryption oracle \$ and the quantum ideal cipher oracle $|E^{\pm}\rangle$ until $\mathcal{A}$ *receives* the $(z+1)$th block in the response to the encryption queries. When $\mathcal{A}$ receives the $(z+1)$th block, sample u keys $\mathbf{K} = (K_1, \ldots, K_u)$ from $\{0,1\}^k$ uniformly at random, replace the oracle \$ (resp., the quantum oracles of E and its inverse) with $\mathsf{Enc}^{E[\![\mathsf{T}_{z+1}^{\mathsf{enc}}, \mathbf{K}]\!]}$ (resp., the quantum oracles of $E[\![\mathsf{T}_{z+1}^{\mathsf{enc}}, \mathbf{K}]\!]$ and its inverse), and continue running $\mathcal{A}$ until it returns the final output.

The difference between $\mathbf{H}_1^z$ and $\mathbf{H}_2^z$ is bounded as follows.

Lemma 5. $|\Pr[1 \leftarrow \mathcal{A}\ in\ \mathbf{H}_1^z] - \Pr[1 \leftarrow \mathcal{A}\ in\ \mathbf{H}_2^z]| \leq 4\sqrt{p \cdot 2^{-k}}$.

Proof. We prove the lemma by using the resampling lemma (Lemma 2). Let $\mathcal{A}'$ be a distinguisher for the game of Lemma 2 defined as follows.

1-0. The distinguisher is given quantum oracle access to E and E^{-1}.
1-1. The distinguisher $\mathcal{A}'$ runs $\mathcal{A}$, simulating the classical encryption oracle by \$ and forwarding quantum queries to E/E^{-1}.
1-2. As in $\mathbf{H}_1^z$, pause $\mathcal{A}$ just before the $(z+1)$th block (counted from when $\mathcal{A}$ started) is sampled in the responses to the classical encryption queries. Suppose that this block is the wth block in the response to the classical encryption query (id, N, AD, M), and let $Y := \mathsf{pad}(N)$ and ℓ is the block length of M.
1-3. Define a distribution D over $\{0,1\}^k \times \{0,1\}^n \times \{0,1\}^n$ as $D(K_*, s_0, s_1) := \delta_{Y+w,s_0}/2^{k+n}$ if $w \leq \ell$, and $D(K_*, s_0, s_1) := \delta_{Y,s_0}/2^{k+n}$ if $w = \ell + 1$. Return D as the output of $\mathcal{A}'$ in the first stage.
2-0. (K_*, s_0, s_1) is sampled according to D, and a bit b is chosen uniformly at random. The distinguisher $\mathcal{A}'$ is given the tuple (K_*, s_0, s_1), and the quantum oracle access to $E^{(0)} := E$ if $b = 0$ (resp., $E^{(1)} := E \circ \mathsf{swap}_{s_0,s_1}^{K_*}$ if $b = 1$).
2-1. $\mathcal{A}'$ sets $K_{\mathsf{id}} := K_*$ (note that $\mathsf{id} \in \{1, \ldots, u\}$), randomly samples $K_1, \ldots, K_u$ from $\{0,1\}^k$ except for K_{id}, and set $\mathbf{K} = (K_1, \ldots, K_u)$. $\mathcal{A}'$ resumes running $\mathcal{A}$, simulating the classical encryption oracle by $\mathsf{Enc}^{E^{(b)}[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]}$ and quantum primitive oracles by $E^{(b)}[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]$ and its inverse. Finally, $\mathcal{A}'$ outputs the output of $\mathcal{A}$ as its own output.

Then, $\mathcal{A}'$ perfectly simulates $\mathbf{H}_1^z$ and $\mathbf{H}_2^z$ when $b = 0$ and $b = 1$, respectively, and at most p primitive queries are made to E in the first stage of $\mathcal{A}'$. Hence, from Lemma 2, it follows that

$$\begin{aligned}
&|\Pr[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^z] - \Pr[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_2^z]| \\
&\quad \leq |\Pr[1 \leftarrow \mathcal{A}' | b = 0] - \Pr[1 \leftarrow \mathcal{A}' | b = 1]| \\
&\quad \leq 4\sqrt{p \cdot 2^{n - H_\infty(D)}} = 4\sqrt{p \cdot 2^{-k}}.
\end{aligned}$$

□

The difference between $\mathbf{H}_2^z$ and $\mathbf{H}_3^z$ is bounded as follows.

Lemma 6. $|\Pr[1 \leftarrow \mathcal{A}\ in\ \mathbf{H}_2^z] - \Pr[1 \leftarrow \mathcal{A}\ in\ \mathbf{H}_3^z]| \leq \frac{8\ell_{\max}^H}{2^{(\ell_{\max}^H - 2)n/2}} + \frac{1}{2^{2n}} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}} + \frac{(n\ell_{\max}^H + 8)Q_e + (n+8)B_e + 2z + 2}{2^{n+1}} + \frac{Q_e + B_e + 2d}{2^k}$.

Proof. Even if the sampling of the $(z+1)$th block and s_1 in $\mathbf{H}_3^z$ are modified as below, their distribution does not change.

(i) Sample $s_1 \in \{0,1\}^n$ uniformly at random.
(ii) Set the $(z+1)$th block as $C[w] := \bar{M}[w] \oplus E_{K^\star}(s_1)$ for $w \leq \ell$, and $T := H_L(AD, \tilde{C}) \oplus E_{K^\star}(s_1)$ for $w := \ell + 1$.

Here, $\tilde{C}$ denotes the truncated version of C such that $|\tilde{C}| = |M|$, and $K^\star$ is the key involved in the $(z+1)$th block. Thus, in what follows, we regard that $\mathbf{H}_3^z$ samples the $(z+1)$th block and s_1 in this way.

Let[5] $i^\star := \mathsf{id} \in \{1, \ldots, u\}$ be the key index involved in the $(z+1)$th block, and

$$x^\star := \begin{cases} Y + w & \text{if } w \leq \ell, \\ Y & \text{if } w = \ell + 1 \end{cases}$$

(note that $K^\star := K_{i^\star}$ holds). For $j = 1, \ldots, z+1$ and $L \in \{0,1\}^n$, we define $W_j(L) \in \{0,1\}^n$ by

$$W_j(L) := \bar{M}[w] \oplus C[w]$$

if the jth block in the responses (counted from the beginning) is *not* the last (tag) block, and it is the wth block in a classical encryption query ($\bar{M}[w]$ and $C[w]$ are wth block of the message and the ciphertext involved in this classical query, respectively), and

$$W_j(L) := T \oplus H_L(AD, \tilde{C})$$

if the jth block in the responses is the last (tag) block in a classical encryption query, where the query is and the response are denoted by (id, N, AD, M) and (C, T), respectively. (Note that L is not necessarily equal to $E_{K_{\mathsf{id}}}(0)$, and $W_j(L)$ is in fact independent from L if the jth block is not the last (tag) block). In addition, let $\mathsf{id}(j)$ be the index in $\{1, \ldots, u\}$ such that the W_j is defined with respect to the classical query of the form $(\mathsf{id}(j), N, AD, M)$ for some N, AD, M. We also define

$$\widetilde{x}_j := E^{-1}_{K_{\mathsf{id}(j)}}\big(W_j(E_{K_{\mathsf{id}(j)}}(0))\big).$$

In addition, let $Y_j := \mathsf{pad}(N)$ and

$$x_j := \begin{cases} Y_j + w & \text{if the } j\text{th block is not the tag block,} \\ Y_j & \text{if the } j\text{th block is the tag block.} \end{cases}$$

Define bad to be the event that at least one of the following conditions holds when s_1 and the $(z+1)$th blocks are defined:

1. $(K^\star, x^\star) = (K_{\mathsf{id}(j)}, x_j)$ for some $j \leq z$;
2. $(K^\star, x^\star) = (K_{\mathsf{id}(j)}, \widetilde{x}_j)$ for some $j \leq z$;
3. $(K^\star, s_1) = (K_{\mathsf{id}(j)}, x_j)$ for some $j \leq z$;
4. $(K^\star, s_1) = (K_{\mathsf{id}(j)}, \widetilde{x}_j)$ for some $j \leq z$;
5. $s_1 = 0$.

[5] From this point on, we define the bad event bad and bound its probability. The authors developed the high-level idea underlying the definition of bad, while the technical details of the argument were worked out with the assistance of ChatGPT through multiple rounds of interaction, including feedback, corrections, and refinements by the authors. The authors independently checked the resulting argument for correctness and take full responsibility for its contents.

6. $\tilde{x}_j = 0$ for some $j \leq z$.

(When defining $E'[\![\mathsf{T}_z^{\mathsf{enc}}, \boldsymbol{K}]\!]$ or $E[\![\mathsf{T}_{z+1}^{\mathsf{enc}}, \mathbf{K}]\!]$ from E, the values of x_j and $\widetilde{x}_j$ are swapped for each j w.r.t the key $K_{\mathsf{id}(j)}$. If bad does not happen, then the reprogrammed values do not overlap with each other, nor with 0.) For $i = 1, \ldots, 6$, let bad_i be the event that bad happens and the ith condition among the six conditions holds.

Then, the probability of bad is the same between $\mathbf{H}_2^z$ and $\mathbf{H}_3^z$, and the behavior of the oracles are identical between the two hybrids as long as bad does not happen. Thus,

$$|\Pr[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_2^z] - \Pr[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_3^z]| \leq \Pr[\mathsf{bad}] \tag{6}$$

holds.

As for bad_1, we have

$$\Pr[\mathsf{bad}_1 \wedge i^\star \neq \mathsf{id}(j)] \leq \frac{d}{2^k}, \quad \Pr[\mathsf{bad}_1 \wedge i^\star = \mathsf{id}(j)] = 0,$$

because $\mathcal{A}$ is nonce-respecting, and $\mathcal{A}$ queries every nonce to at most d keys. Hence,

$$\Pr[\mathsf{bad}_1] \leq \frac{d}{2^k}. \tag{7}$$

Next, we bound $\Pr[\mathsf{bad}_2]$. Let

$$\Lambda_1 := \left\lceil \frac{\ell_{\max}^H n}{2}\, 2^{k-n} \right\rceil, \quad \Lambda_2 := \left\lceil n 2^{k-n} \right\rceil.$$

Define[6] load_1 to be the event that

$$\left|\left\{K \in \{0,1\}^k : E_K(x) \oplus H_{E_K(0)}(X) = t\right\}\right| \geq \Lambda_1$$

for some $x \neq 0$, $t \in \{0,1\}^n$, and some hash input $X = (AD, \tilde{C})$ of the block length at most $\ell_{\max}^H$, and load_2 be the event that

$$\left|\left\{K \in \{0,1\}^k : E_K(x) = t\right\}\right| \geq \Lambda_2,$$

for some $x \neq 0$ and $t \in \{0,1\}^n$.

For fixed $x \neq 0$ and X, view each key K as a ball placed in the bin

$$E_K(x) \oplus H_{E_K(0)}(X).$$

Conditioned on the preceding balls, the probability that the next ball falls into any fixed bin is at most

$$\frac{1}{2^n - 1} \leq 2^{1-n}.$$

[6] The events load_1 and load_2 are defined based on balls-into-bins arguments in [HTT18].

Hence, by applying Lemma 3 (with $q = 2^k$, $a = 2^{k-n}$, and $\ell = \ell^H_{\max}$), the probability that the maximum load is at least Λ_1 is upper bounded by $2^{-1.5\ell^H_{\max} n}$. Since the number of all possible $X = (AD, \tilde{C})$ is at most

$$\begin{aligned}\#\left\{(AD,\tilde{C}) : |AD|_n + |\tilde{C}|_n \le \ell^H_{\max}\right\} &\le \sum_{\substack{a,b\ge 1,\\ a+b\le \ell^H_{\max}}} 2^{an+1}2^{bn+1} = 4\sum_{2\le w\le \ell^H_{\max}} (w-1)2^{wn} \\ &\le 8\ell^H_{\max} 2^{\ell^H_{\max} n}, \end{aligned} \tag{8}$$

taking a union bound over all possible x and X gives

$$\Pr\left[\mathsf{load}_1\right] \le 2^n \cdot \left(8\ell^H_{\max}2^{\ell^H_{\max}n}\right) \cdot 2^{-1.5\ell^H_{\max}n} = \ell^H_{\max}2^{n-\ell^H_{\max}n/2+3}.$$

We can also show

$$\Pr\left[\mathsf{load}_2\right] \le 2^n \cdot 2^{-3n} = 2^{-2n}$$

in the same way (by applying Lemma 3 with $q = 2^k$, $a = 2^{k-n}$, and $\ell = 2$, and then taking the union bound for x).

We also define mcoll to be the event that

$$W_{j_1}(L) = \cdots = W_{j_{d+1}}(L)$$

for some $L \in \{0,1\}^n$ and $1 \le j_1 < \cdots < j_{d+1} \le z$. Since each response block of the ideal encryption oracle is freshly and uniformly sampled,

$$\Pr[\mathsf{mcoll}] \le 2^n \binom{z}{d+1} 2^{-dn} \le \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}}.$$

Next, we bound the probability of bad_2 conditioned on $\neg\mathsf{load}_1$, $\neg\mathsf{load}_2$, and $\neg\mathsf{mcoll}$. For the case when $\mathsf{id}(j) = i^\star$ and the jth block is the tag block, conditioned on $\neg\mathsf{load}_1$, we have

$$\begin{aligned}&\Pr\left[\mathsf{bad}_2 \wedge \mathsf{id}(j) = i^\star \wedge j\text{th block is the tag block} | \neg\mathsf{load}_1\right] \\ &\quad= \Pr\left[\exists j \le z :\ \mathsf{id}(j) = i^\star \wedge x^\star = \widetilde{x}_j \wedge j\text{th block is the tag block} \,\middle|\, \neg\mathsf{load}_1\right] \\ &\quad\le \sum_{\substack{j:\mathsf{id}(j)=i^\star \\ j\text{th block is the tag block}}} \Pr\left[E_{K_{i^\star}}(x^\star) = W_j(E_{K_{i^\star}}(0)) | \neg\mathsf{load}_1\right] \\ &\quad\le \sum_{\substack{j:\mathsf{id}(j)=i^\star \\ j\text{th block is the tag block}}} \frac{\Lambda_1}{2^k} \le \frac{Q_e\Lambda_1}{2^k}. \end{aligned} \tag{9}$$

For the case when $\mathsf{id}(j) = i^\star$ and the jth block is *not* the tag block, conditioned on $\neg\mathsf{load}_2$, we have

$$\begin{aligned}
&\Pr\left[\mathsf{bad}_2 \wedge \mathsf{id}(j) = i^\star \wedge j\text{th block is not the tag block} | \neg\mathsf{load}_2\right] \\
&\quad = \Pr\left[\exists j \le z:\ \mathsf{id}(j) = i^\star \wedge x^\star = \widetilde{x}_j \wedge j\text{th block is not the tag block} \,\middle|\, \neg\mathsf{load}_2\right] \\
&\quad \le \sum_{\substack{j:\mathsf{id}(j)=i^\star \\ j\text{th block is not the tag block}}} \Pr\left[E_{K_{i^\star}}(x^\star) = W_j(E_{K_{i^\star}}(0)) | \neg\mathsf{load}_2\right] \\
&\quad \le \sum_{\substack{j:\mathsf{id}(j)=i^\star \\ j\text{th block is not the tag block}}} \frac{\Lambda_2}{2^k} \le \frac{B_e\Lambda_2}{2^k}. \qquad (10)
\end{aligned}$$

For the case of $\mathsf{id}(j) \neq i^\star$ conditioned on $\neg\mathsf{mcoll}$, we have

$$\begin{aligned}
&\Pr\left[\mathsf{bad}_2 \wedge \mathsf{id}(j) \neq i^\star | \neg\mathsf{mcoll}\right] \\
&\quad = \Pr\left[\mathsf{id}(j) \neq i^\star \wedge K_{i^\star} = K_{\mathsf{id}(j)} \wedge x^\star = \widetilde{x}_j \middle| \neg\mathsf{mcoll}\right] \\
&\quad \le \frac{d}{2^k}. \qquad (11)
\end{aligned}$$

Combining the above bounds, we obtain

$$\begin{aligned}
\Pr[\mathsf{bad}_2] &\le \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = i^\star] + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) \neq i^\star] \\
&\le \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = i^\star \wedge j\text{th block is the tag block}] \\
&\quad + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = i^\star \wedge j\text{th block is not the tag block}] \\
&\quad + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) \neq i^\star] \\
&\le \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = i^\star \wedge j\text{th block is the tag block} | \neg\mathsf{load}_1] \\
&\quad + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = i^\star \wedge j\text{th block is not the tag block} | \neg\mathsf{load}_2] \\
&\quad + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) \neq i^\star] \\
&\quad + \Pr[\mathsf{load}_1] + \Pr[\mathsf{load}_2] + \Pr[\mathsf{mcoll}] \\
&\le \frac{Q_e\Lambda_1}{2^k} + \frac{B_e\Lambda_2}{2^k} + \frac{d}{2^k} + \ell_{\max}^H 2^{n-\ell_{\max}^H n/2+3} + 2^{-2n} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}}. \qquad (12)
\end{aligned}$$

Since

$$\frac{\Lambda_1}{2^k} \le \frac{\ell_{\max}^H n}{2^{n+1}} + \frac{1}{2^k} \text{ and } \frac{\Lambda_2}{2^k} \le \frac{n}{2^{n+1}} + \frac{1}{2^k},$$

it follows that

$$\Pr[\mathsf{bad}_2] \le \frac{8\ell_{\max}^H}{2^{(\ell_{\max}^H-2)n/2}} + \frac{1}{2^{2n}} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}} + \frac{n\ell_{\max}^H Q_e + nB_e}{2^{n+1}} + \frac{Q_e + B_e + d}{2^k}. \qquad (13)$$

As for bad_3, we have

$$\begin{aligned}\Pr[\mathsf{bad}_3] &\le \Pr[\mathsf{bad}_3 \wedge \mathsf{id}(j) = i^\star] + \Pr[\mathsf{bad}_3 \wedge \mathsf{id}(j) \neq i^\star] \\ &\le \frac{Q_e + B_e}{2^n} + \frac{u-1}{2^k} \cdot \frac{Q_e + B_e}{2^n} \\ &\le \frac{2(Q_e + B_e)}{2^n} \end{aligned} \tag{14}$$

because the keys and s_1 are chosen uniformly at random, and $u \le 2^k$. Similarly,

$$\Pr[\mathsf{bad}_4] \le \frac{2(Q_e + B_e)}{2^n} \tag{15}$$

and

$$\Pr[\mathsf{bad}_5] \le \frac{1}{2^n} \tag{16}$$

hold.

Furthermore, $\Pr[\tilde{x}_j = 0] = 1/2^n$ since $W_j(L)$ is uniformly distributed over $\{0,1\}^n$ for any L and $j \le z+1$ (when E and the keys are fixed). Thus,

$$\Pr[\mathsf{bad}_6] \le \frac{z}{2^n}. \tag{17}$$

By Eq (7), (13), and (14)-(17), we have

$$\begin{aligned}\Pr[\mathsf{bad}] \le{}& \frac{8\ell_{\max}^H}{2^{(\ell_{\max}^H - 2)n/2}} + \frac{1}{2^{2n}} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}} \\ &+ \frac{\left(n\ell_{\max}^H + 8\right) Q_e + (n+8)\, B_e + 2z + 2}{2^{n+1}} + \frac{Q_e + B_e + 2d}{2^k}. \end{aligned} \tag{18}$$

The claim of the lemma follows from this inequality and Eq. (6). □

The difference between $\mathbf{H}_3^z$ and $\mathbf{H}_4^z$ is bounded as follows.

Lemma 7. $|\Pr[1 \leftarrow \mathcal{A}\ in\ \mathbf{H}_3^z] - \Pr[1 \leftarrow \mathcal{A}\ in\ \mathbf{H}_4^z]| \le \frac{8\ell_{\max}^H}{2^{(\ell_{\max}^H - 2)n/2}} + \frac{1}{2^{2n}} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}} + \frac{\left(n\ell_{\max}^H + 8\right)Q_e + (n+8)B_e + 2z + 2}{2^{n+1}} + \frac{Q_e + B_e + 2d}{2^k}$.

Proof. Let bad be the event as defined in the proof of Lemma 6. Then, the oracles of $\mathbf{H}_3^z$ and $\mathbf{H}_4^z$ are perfectly indistinguishable if bad does not happen. Hence, the difference between the two hybrids is bounded by $\Pr[\mathsf{bad}]$, which is further bounded as in Eq. (18). Thus, the claim of the lemma holds. □

The difference between $\mathbf{H}_4^z$ and $\mathbf{H}_1^{z+1}$ is bounded as follows.

Lemma 8. *Let p_z be the expected value of the number of quantum primitive queries to E/E^{-1} that $\mathcal{A}$ makes in the ideal world after receiving the $(z+1)$th block (counted from the beginning of the game) and before receiving the $(z+2)$th block (or, before $\mathcal{A}$ stops if the $(z+1)$th block is the final block received) in the responses to the classical encryption queries. Then,*

$$\left|\Pr[1 \leftarrow \mathcal{A}\ in\ \mathbf{H}_4^z] - \Pr\left[1 \leftarrow \mathcal{A}\ in\ \mathbf{H}_1^{z+1}\right]\right| \le 2p_z\sqrt{\frac{4d}{2^k}} + \frac{2(z+1)^{d+1}}{(d+1)!2^{(d-1)n}}. \tag{19}$$

Proof. For $j = 1, \ldots, z+1$ and $L \in \{0,1\}^n$, we define $W_j(L) \in \{0,1\}^n$ by

$$W_j(L) := \bar{M}[w] \oplus C[w]$$

if the jth block in the responses (counted from the beginning) is *not* the last (tag) block, and it is the wth block in a classical encryption query ($\bar{M}[w]$ and $C[w]$ are wth block of the (zero-padded) message and the ciphertext involved in this classical encryption query, respectively), and

$$W_j(L) := T \oplus H_L(AD, \tilde{C})$$

if the jth block in the responses is the last (tag) block in a classical encryption query, where the query is and the response are denoted by (id, N, AD, M) and (C, T), respectively. (As before, $\tilde{C}$ is the truncated version of C such that $|\tilde{C}| = |M|$. Note that L is not necessarily equal to $E_{K_{\mathsf{id}}}(0)$, and $W_j(L)$ is in fact independent from L if the jth block is not the last (tag) block). In addition, let $\mathsf{id}(j)$ be the index in $\{1, \ldots, u\}$ such that the W_j is defined with respect to the classical query of the form $(\mathsf{id}(j), N, AD, M)$ for some N, AD, M. Next, define[7] mcoll to be the event that $W_{j_1}(L) = W_{j_2}(L) = \cdots = W_{j_{d+1}}(L)$ holds for some $1 \le j_1 < \cdots < j_{d+1} \le z+1$ and some $L \in \{0,1\}^n$. Then, since the responses (C, T) from the ideal encryption oracle \$ are always freshly sampled random strings, we have

$$\begin{aligned}
\Pr[\mathsf{mcoll}] &\le \sum_{L \in \{0,1\}^n} \sum_{1 \le j_1 < \cdots < j_{d+1} \le z+1} \Pr\left[W_{j_1}(L) = \cdots = W_{j_{d+1}}(L)\right] \\
&\le 2^n \cdot \binom{z+1}{d+1} \cdot \left(\frac{1}{2^n}\right)^d = \frac{(z+1)z(z-1)\cdots(z-d+1)}{(d+1)!} \frac{1}{2^{(d-1)n}} \\
&\le \frac{(z+1)^{d+1}}{(d+1)! 2^{(d-1)n}}. \qquad (20)
\end{aligned}$$

In what follows, we prove Eq. (19) by using the arbitrary reprogramming lemma (Lemma 1). Let $\mathcal{A}'$ be a distinguisher for the game of Lemma 1 defined as follows.

1-1. $\mathcal{A}'$ samples an ideal cipher E by itself and runs $\mathcal{A}$. When $\mathcal{A}$ makes classical encryption queries (resp., quantum primitive queries), $\mathcal{A}'$ responds by simulating \$ (resp., by using E). As in $\mathbf{H}_4^z$, pause $\mathcal{A}$ when it receives the $(z+1)$th block (counted from when $\mathcal{A}$ started) in the response to the encryption queries. Check if mcoll has happened or not.

1-2. Define $F_0 : \{0,1\} \times \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ as $F_0(a, K, x) = E_K^{1-2a}(x)$. Let $\mathcal{B}$ be an algorithm that samples random u keys $\mathbf{K} := (K_1, \ldots, K_u)$, and then return the function $F_1 : \{0,1\} \times \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ defined

[7] The idea of introducing the event mcoll, as well as the essential ideas underlying the proofs of Eq. (23) and Eq. (24), were suggested by ChatGPT. The authors subsequently worked out and verified the details of the arguments and take full responsibility for their correctness.

by $F_1(a,K,x) := \left(E[\![\mathsf{T}^{\mathsf{enc}}_{z+1}, \mathbf{K}]\!]\right)^{1-2a}(x)$ (where $\mathsf{T}^{\mathsf{enc}}_{z+1}$ is the transcript of the encryption queries by $\mathcal{A}$ and the responses defined so far) if mcoll has not happened, and $F_1 := F_0$ if mcoll has happened. $\mathcal{A}'$ returns F_0 and $\mathcal{B}$ as the output of the first stage.

2-1. Run $\mathcal{B}$ to sample $\mathbf{K} = (K_1, \ldots, K_u)$ and define F_1. A random bit b is sampled.

2-2. $\mathcal{A}'$ is given quantum oracle access to F_b. $\mathcal{A}'$ continues running $\mathcal{A}$. If the $(z+1)$th block is *not* the final block (tag block) in the current response to the classical encryption query, do nothing and just proceed to the next step. If the $(z+1)$th block is the final block (tag block), then continue running $\mathcal{A}$ (in this case, the next queries by $\mathcal{A}$ are quantum primitive queries). When $\mathcal{A}$ makes quantum primitive queries, $\mathcal{A}'$ simulates the quantum primitive oracles by using $F_b(0,\cdot,\cdot)$ and $F_b(1,\cdot,\cdot)$, and pauses $\mathcal{A}$ when it makes the next classical encryption query. Then, proceeds to the next step.

3-1. $\mathcal{A}'$ is given the randomness to define F_1, i.e., the keys $\mathbf{K} = (K_1, \ldots, K_u)$. $\mathcal{A}'$ resumes running $\mathcal{A}$, simulating the classical encryption oracle by $\mathsf{Enc}^{E[\![\mathsf{T}^{\mathsf{enc}}_{z+1}, \mathbf{K}]\!]}$ and the quantum primitive oracle by $E[\![\mathsf{T}^{\mathsf{enc}}_{z+1}, \mathbf{K}]\!]$.

3-2. When $\mathcal{A}$ returns an output, $\mathcal{A}'$ returns it as its own output.

When $b = 0$, the above $\mathcal{A}'$ perfectly simulates $\mathbf{H}_1^{z+1}$, and the expected value of the number of quantum queries by $\mathcal{A}'$ made in the second stage is p_z. Moreover, when $b = 1$ and mcoll does not happen, then $\mathcal{A}'$ also perfectly simulates $\mathbf{H}_4^z$. In particular, we have

$$\begin{aligned}
&\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \\
&\quad \le \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z | \neg\mathsf{mcoll}\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| + \Pr[\mathsf{mcoll}] \\
&\quad \le \left|\Pr\left[1 \leftarrow \mathcal{A}' | b = 1 \wedge \neg\mathsf{mcoll}\right] - \Pr\left[1 \leftarrow \mathcal{A}' | b = 0\right]\right| + \Pr[\mathsf{mcoll}] \\
&\quad \le \left|\Pr\left[1 \leftarrow \mathcal{A}' | b = 1\right] - \Pr\left[1 \leftarrow \mathcal{A}' | b = 0\right]\right| + 2\Pr[\mathsf{mcoll}]. \qquad (21)
\end{aligned}$$

In what follows, we bound the parameter ε of Lemma 1.

Fix $(a,K,x) \in \{0,1\} \times \{0,1\}^k \times \{0,1\}^n$ arbitrarily. If $F_0(a,K,x) \neq F_1(a,K,x)$, then this (a,K,x) must be one of the values on which E/E^{-1} is reprogrammed when defining $E[\![\mathsf{T}^{\mathsf{enc}}_{z+1}, \mathbf{K}]\!]$ (see Eq. (5)), i.e., (a,K,x) satisfies either of the following four conditions.

(i) $a = 0$ and there is a previous classical encryption query (id, N, AD, M) such that $K = K_{\mathsf{id}}$ and $x = Y + w$ for some $0 \le w \le \ell$, where $Y = \mathsf{pad}(N)$ and ℓ is the block length of M.

(ii) $a = 1$ and there is a previous classical encryption query (id, N, AD, M) such that $K = K_{\mathsf{id}}$ and $x = E_K(Y + w)$ for some $0 \le w \le \ell$.

(iii) $a = 0$ and there is a previous classical encryption query (id, N, AD, M) and the response (C,T) (which may be only partially defined) such that $K = K_{\mathsf{id}}$, and $x = E_K^{-1}(\bar{M}[w] \oplus C[w])$ for some $1 \le w \le \ell$, or $x = E_K^{-1}(T \oplus H_L(AD, \tilde{C}))$, where $L := E_{K_{\mathsf{id}}}(0)$.

(iv) $a = 1$ and there is a previous classical encryption query (id, N, AD, M) and the response (C,T) (which may only partially defined) such that $K = K_{\mathsf{id}}$, and $x = \bar{M}[w] \oplus C[w]$ for some $1 \le w \le \ell$, or $x = T \oplus H_L(AD, \tilde{C})$, where $L := E_{K_{\mathsf{id}}}(0)$.

Recall that (a, K, x) is fixed. For the case of (i), the probability of $F_1(a, K, x) \neq F_0(a, K, x)$ under the random choice of $K_1, \ldots, K_u$ is bounded as

$$\Pr_{K_1,\ldots,K_u}[F_1(a,K,x) \neq F_0(a,K,x) \wedge \text{Case (i)}] \leq \frac{d}{2^k} \tag{22}$$

because $\mathcal{A}$ queries every single nonce to the encryption oracle for at most d keys.

As for the case of (ii), let $I(\mathsf{T}_z^{\mathsf{enc}})$ be the subset of the indices id in $\{1, \ldots, u\}$ such that (id, N, AD, M) appears in $\mathsf{T}_z^{\mathsf{enc}}$ for some (N, AD, M), and there is $w \in \{0, \ldots, \ell\}$ ($\ell := \lceil |M|/n \rceil$) such that $E_K^{-1}(x) = \mathsf{pad}(N) + w$ (recall again that now (a, K, x) is fixed, and thus $E_K^{-1}(x)$ is also a fixed value here). Then, we have

$$\begin{aligned}
&\Pr_{K_1,\ldots,K_u}[F_1(a,K,x) \neq F_0(a,K,x) \wedge \text{Case (ii)}] \\
&\quad \leq \Pr_{K_1,\ldots,K_u}\left[K = K_{\mathsf{id}} \text{ for some } \mathsf{id} \in I(\mathsf{T}_{z+1}^{\mathsf{enc}})\right] \\
&\quad \leq \sum_{\mathsf{id} \in I(\mathsf{T}_{z+1}^{\mathsf{enc}})} \Pr_{K_{\mathsf{id}}}[K = K_{\mathsf{id}}] = \frac{\left|I(\mathsf{T}_{z+1}^{\mathsf{enc}})\right|}{2^k} \leq \frac{d}{2^k},
\end{aligned} \tag{23}$$

where the last inequality follows from the assumption that $\mathcal{A}$ queries every single nonce to the encryption oracle for at most d keys (i.e., at most d indices id).

Next, we analyze the case of (iii). Since $F_1 = F_0$ if mcoll happens, and $|\{j : W_j(L) = y\}| \leq d$ for any $L \in \{0,1\}^n$ if mcoll does not happen, we have

$$\begin{aligned}
&\Pr_{K_1,\ldots,K_u}[F_1(a,K,x) \neq F_0(a,K,x) \wedge \text{Case (iii)}] \\
&\quad \leq \Pr_{K_1,\ldots,K_u}\left[K = K_{\mathsf{id}(j)} \wedge E_K(x) = W_j(E_K(0)) \text{ for some } j \in \{1, \ldots, z+1\} \middle| \neg\mathsf{mcoll}\right] \\
&\quad \leq \sum_{j : W_j(E_K(0)) = E_K(x)} \Pr_{K_{\mathsf{id}(j)}}\left[K = K_{\mathsf{id}(j)}\right] \\
&\quad = \frac{|\{j : W_j(E_K(0)) = E_K(x)\}|}{2^k} \leq \frac{d}{2^k}.
\end{aligned} \tag{24}$$

In addition,

$$\Pr_{K_1,\ldots,K_u}[F_1(a,K,x) \neq F_0(a,K,x) \wedge \text{Case (iv)}] \leq \frac{d}{2^k} \tag{25}$$

can be shown analogously.

By using Eq. (22), (23), (24), and (25), the value ε of Lemma 1 can be upper bounded as

$$\varepsilon = \max_{a,K,x} \Pr_{K_1,\ldots,K_u}[F_1(a,K,x) \neq F_0(a,K,x)] \leq \frac{4d}{2^k}. \tag{26}$$

Thus, from Eq. (21),

$$\begin{aligned}
&\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \\
&\quad \leq \left|\Pr\left[1 \leftarrow \mathcal{A}' | b = 1\right] - \Pr\left[1 \leftarrow \mathcal{A}' | b = 0\right]\right| \\
&\quad \leq 2p_z\sqrt{\frac{4d}{2^k}} + \frac{2(z+1)^{d+1}}{(d+1)!2^{(d-1)n}}
\end{aligned}$$

follows. □

Finishing the Proof of Theorem 1. For each $z \in \{0, 1, \ldots, \sigma_e + q_e - 1\}$ we have

$$\begin{aligned}
&\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^0\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{\sigma_e+q_e}\right]\right| \\
&\quad \leq \sum_{0 \leq z < \sigma_e+q_e} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \\
&\quad \leq \sum_{\substack{0 \leq z < \sigma_e+q_e \\ i=1,2,3}} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_i^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_{i+1}^z\right]\right| \\
&\qquad + \sum_{0 \leq z < \sigma_e+q_e} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \qquad (27)
\end{aligned}$$

In addition,

$$\sum_{0 \leq z < \sigma_e+q_e} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_2^z\right]\right| \leq 4(\sigma_e + q_e)\sqrt{\frac{p}{2^k}} \qquad (28)$$

follows from Lemma 5,

$$\begin{aligned}
&\sum_{\substack{0 \leq z < \sigma_e+q_e \\ i=2,3}} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_i^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_{i+1}^z\right]\right| \\
&\quad \leq \frac{16(\sigma_e + q_e)\ell_{\max}^H}{2^{(\ell_{\max}^H - 2)n/2}} + \frac{2(\sigma_e + q_e)}{2^{2n}} + \frac{2(\sigma_e + q_e)^{d+2}}{(d+1)!\, 2^{(d-1)n}} \\
&\qquad + \frac{2\left(n\ell_{\max}^H + 8\right)(\sigma_e + q_e)Q_e + 2\,(n+8)\,(\sigma_e + q_e)B_e + 4(\sigma_e + q_e + 1)^2}{2^{n+1}} \\
&\qquad + \frac{2(\sigma_e + q_e)(Q_e + B_e + 2d)}{2^k} \qquad (29)
\end{aligned}$$

follows from Lemma 6 and Lemma 7, and

$$\begin{aligned}
&\sum_{0 \leq z < \sigma_e+q_e} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \\
&\quad \leq 2p\sqrt{\frac{4d}{2^k}} + \frac{2(\sigma_e + q_e + 1)^{d+2}}{(d+1)!2^{(d-1)n}} \qquad (30)
\end{aligned}$$

follows from Lemma 8. From Eq. (27)-(30), we obtain

$$\begin{aligned}
&\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^0\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{\sigma_e+q_e}\right]\right| \\
&\quad \le 4(\sigma_e+q_e)\sqrt{\frac{p}{2^k}} + 2p\sqrt{\frac{4d}{2^k}} \\
&\qquad + \frac{16(\sigma_e+q_e)\ell_{\max}^H}{2^{(\ell_{\max}^H-2)n/2}} + \frac{2(\sigma_e+q_e)}{2^{2n}} + \frac{4(\sigma_e+q_e+1)^{d+2}}{(d+1)!\,2^{(d-1)n}} \\
&\qquad + \frac{2\left(n\ell_{\max}^H+8\right)(\sigma_e+q_e)Q_e + 2\left(n+8\right)(\sigma_e+q_e)B_e + 4(\sigma_e+q_e+1)^2}{2^{n+1}} \\
&\qquad + \frac{2(\sigma_e+q_e)(Q_e+B_e+2d)}{2^k}
\end{aligned} \tag{31}$$

Since the real and ideal worlds (with the modified oracles as defined below Theorem 1) correspond to $\mathbf{H}_1^0$ and $\mathbf{H}_1^{\sigma_e+q_e}$, respectively, the claim of Theorem 1 follows.

5 Post-Quantum Authenticity of CAU in the QICM

As for the post-quantum authenticity of CAU, the following theorem holds.

Theorem 2. *Suppose $u \le 2^k$, and H is a c-AXU. Suppose that $\mathcal{A}$ makes at most q_e classical encryption queries such that the length of the messages is at most σ_e blocks in total, $\mathcal{A}$ makes at most q_d classical verification queries, and $\mathcal{A}$ queries every single nonce to the encryption oracle for at most d keys. Additionally, suppose that $\mathcal{A}$ makes at most Q_e classical encryption queries for each key, the length of messages for each key is at most B_e blocks in total, and that the block length of (AD, C) (input to H) is at most $\ell_{\max}^H$ for every encryption/verification query. Moreover, assume that the number of quantum queries by $\mathcal{A}$ to the ideal cipher E and the inverse E^{-1} is at most p. Then, we have $\mathsf{Adv}_{\mathsf{CAU}}^{\mathsf{mk\text{-}auth}}(\mathcal{A}) \le \delta_{\mathsf{priv}} + \delta'$, where δ_{priv} is as defined in Eq. (2), and*

$$\begin{aligned}
\delta' := q_d &\left(\frac{nc\ell_{\max}^H}{2^\tau} + \frac{\left(\ell_{\max}^H n+1\right)q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{\ell_{\max}^H n+2}{2^\tau}\right) \\
&+ (\sigma_e+q_e+q_d)\left(\frac{(n\ell_{\max}^H+4)Q_e + (n+5)(B_e+1)}{2^n} + \frac{2d}{2^k}\right) \\
&+ \ell_{\max}^H 2^{-(\ell_{\max}^H-2)n/2+3} + \frac{2}{2^{2n}} + \frac{(\sigma_e+q_e)^{d+1}}{(d+1)!2^{(d-1)n}}.
\end{aligned} \tag{32}$$

Proof Overview. The proof[8] first postpones the evaluation of all verification queries until the end of the game. More precisely, we replace the verification oracle by an oracle that always returns $\perp$, while recording all verification queries made by the adversary. This modification does not change the adversary's view before a successful forgery occurs, and we may therefore equivalently determine at the end of the game whether any of the recorded verification queries would have been accepted.

Once the verification oracle always returns $\perp$, we next replace the encryption oracle with the ideal encryption oracle \$, while the recorded verification queries are eventually evaluated with respect to a block cipher reprogrammed to be compatible with the encryption transcript. This replacement is validated by extending the hybrid argument in the privacy proof of Theorem 1. More precisely, we augment each hybrid in that proof by recording all verification queries and evaluating them only after the adversary terminates, using the block cipher associated with that hybrid. The bounds for all hybrid transitions remain unchanged, and hence the total distinguishing loss is at most δ_{priv}. After this transformation, the encryption transcript consists of independently sampled random strings, and all verification queries are evaluated only at the end with respect to the reprogrammed ideal cipher.

It remains to bound the probability that one of the recorded verification queries is accepted in this final game. For this purpose, we condition on suitable good events extending those used in the privacy proof, which guarantee, in particular, that the points of E modified by different reprogramming operations do not collide and that the relevant key-to-output mappings have bounded maximum load. Under these good events, the forgery probability is bounded using the c-AXU property of H together with counting and maximum-load arguments based on [HTT18].

Modification to the Encryption/Verification Oracles. We modify the encryption oracle as in the proof of Theorem 1. Especially, the length of ciphertexts are always a multiple of n, and we always allow the adversary to see the full n-bit tag. Meanwhile, the verification query and the winning condition for the adversary are not changed: an adversary $\mathcal{A}$ is regarded as having succeeded in forging if a ciphertext and a τ-bit tag pass the verification. Below, we write the details.

1. The (real) encryption oracle is modified as in the proof of Theorem 1: Suppose that a tuple (i, N, AD, M) is queried to the encryption oracle. Each message M is first padded with zeroes at the end so the length will be a multiple of n. The resulting message is denoted by $\bar{M}$ (i.e., $\bar{M} = M||0 \cdots 0$

[8] The authors developed the proof of Theorem 2 with the assistance of ChatGPT, providing the overall direction that reuses the privacy proof. The technical details were worked out through multiple rounds of interaction with ChatGPT, with the authors providing feedback, corrections, and refinements throughout the process. The authors independently checked the resulting argument for correctness and take full responsibility for its contents.

and $|\bar{M}|/n = \lceil |M|/n \rceil$). Then, the ciphertext C is computed for this padded $\bar{M}$, and returned to the adversary. After that, some least significant bits of C are truncated so the length will be equal to $|M|$ (the resulting string is denoted by $\tilde{C}$), the hash value $H_L(AD, \tilde{C})$ is computed, and then the *non-truncated n-bit tag* $T := H_L(AD, \tilde{C}) \oplus E_K(Y)$ is returned as the tag.
2. The verification oracle itself is not changed. The adversary can query some ciphertext $\tilde{C}$ (the length is not necessarily a multiple of n) and a tag $\tilde{T}$ of *length* τ (with a key id i, a nonce N, and associated data AD), and the adversary wins if $\mathsf{Vrf}^E(i, N, AD, \tilde{C}, \tilde{T}) = \top$. To prevent trivial wins, the adversary is prohibited from querying $(i, N, AD, \tilde{C}, \tilde{T})$ to the verification oracle if there is a previous encryption query of the form (i, N, AD, M) for some message M and the response is (C, T) such that $\mathsf{msb}_{|M|}(C) = \tilde{C}$ and $\mathsf{msb}_\tau(T) = \tilde{T}$.

These changes only increase the probability of the adversary forging because the information given to the adversary by the encryption oracle is increased. Hence, in what follows, we assume that the encryption oracle is modified as above.

In particular, note that the length of C (resp., tag T) returned from the encryption oracle is always a multiple of n (resp., equal to n), and the truncated ciphertext given to the hash function H or verification oracle (resp., τ-bit tags given to the verification oracle) is now denoted by $\tilde{C}$ (resp., denoted by $\tilde{T}$). We will also use the symbol $\bar{M}$ to denote the padded message defined as above.

Remark 2. As in the proof of Theorem 1, the oracles' behavior is unchanged if $\tau = n$ and $\mathcal{A}$ queries only messages and ciphertexts the length of which is a multiple of n.

Notations and Terminology. We also reuse other notations and terminology in the proof of Theorem 1. In particular, $\mathsf{T}^{\mathsf{enc}}_z$ denotes the transcript of the classical *encryption* queries and the responses up to the zth block in the responses, counted from the beginning of the game. The reprogrammed cipher $E[\![\mathsf{T}^{\mathsf{enc}}_z, \mathbf{K}]\!]$ is also defined in the same way as before. Additionally, let $\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}$ be the transcript of the classical encryption queries at the end of the game (i.e., $\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}} = \mathsf{T}^{\mathsf{enc}}_{(\sigma_e+q_e)}$), and $\mathsf{T}^{\mathsf{vf}}_{\mathsf{all}}$ be the transcript of all the classical verification queries.

Hybrids. To prove the theorem, we use the following hybrids $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4$.

$\mathbf{H}_1$. This is the real security game that defines authenticity (with the encryption oracle modified as above). The adversary $\mathcal{A}$ is given the (modified) classical encryption oracle Enc^E, the classical verification oracle Vrf^E, and the quantum oracles of the ideal cipher E and its inverse. Set the flag $\mathsf{win} \leftarrow 1$ (which means that $\mathcal{A}$ succeeds to forge) if Vrf^E returns $\top$. (The flag win is initially set to be 0.)

$\mathbf{H}_2$. In this hybrid, the verification oracle is replaced with $\bot$. Still, whenever $\mathcal{A}$ makes a verification query $(i, N, AD, \tilde{C}, \tilde{T})$, the hybrid internally evaluates the

real verification oracle Vrf^E on the query, and sets a flag win as 1 if $\mathsf{Vrf}^E(i, N, AD, \tilde{C}, \tilde{T}) = \top$. The encryption oracle is unchanged.

$\mathbf{H}_3$. In this hybrid, the oracles remain the same as $\mathbf{H}_2$ (the encryption oracle is the real one while the verification oracle is $\perp$), but how to set the flag win is changed: When $\mathcal{A}$ makes a verification query $(i, N, AD, \tilde{C}, \tilde{T})$, the hybrid just records it and does not compute whether it passes the verification. Finally, when $\mathcal{A}$ stops, the hybrid evaluates $\mathsf{Vrf}^E(i, N, AD, \tilde{C}, \tilde{T})$ for all $(i, N, AD, \tilde{C}, \tilde{T}) \in \mathsf{T}^{\mathsf{vf}}_{\mathsf{all}}$, and sets $\mathsf{win} \leftarrow 1$ if Vrf^E returns $\top$ for any of them.

$\mathbf{H}_4$. In this hybrid, the verification oracle is again $\perp$, and the encryption oracle is also replaced with the ideal encryption oracle \$ (as in the proof of Theorem 1, we consider the modified version that always returns a bit string of length multiple of n). How to set the flag win is also changed so that the verification oracle depends on the reprogrammed oracle $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$ instead of E: When $\mathcal{A}$ stops, the hybrid evaluates $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}(i, N, AD, \tilde{C}, \tilde{T})$ for all $(i, N, AD, \tilde{C}, \tilde{T}) \in \mathsf{T}^{\mathsf{vf}}_{\mathsf{all}}$, and sets $\mathsf{win} \leftarrow 1$ if $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}$ returns $\top$ for any of them. Here, we assume that the keys $\mathbf{K} = (K_1, \ldots, K_u)$ are randomly chosen when $\mathcal{A}$ stops.

First, the following lemma holds.

Lemma 9. $\Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_1] = \Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_2] = \Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_3]$.

Proof. The first equation holds because $\mathbf{H}_1$ and $\mathbf{H}_2$ are identical until win is set to be 1. The second equation holds because $\mathcal{A}$'s views are the same between $\mathbf{H}_2$ and $\mathbf{H}_3$. □

The difference between $\mathbf{H}_3$ and $\mathbf{H}_4$ is bounded as follows.

Lemma 10. $|\Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_3] - \Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_4]| \leq \delta_{\mathsf{priv}}$ *holds, where* δ_{priv} *is the value defined as in the statement of Theorem 1.*

Proof. Let $m := \sigma_e + q_e$. We augment each hybrid $\mathbf{H}^z_j$ used in the proof of Theorem 1 ($j \in \{1, 2, 3, 4\}$ and $0 \leq z \leq m$) as follows. In addition to the oracles provided in $\mathbf{H}^z_j$, the adversary $\mathcal{A}$ is given a verification oracle that always returns $\perp$, while all verification queries are recorded in a transcript $\mathsf{T}^{\mathsf{vf}}_{\mathsf{all}}$. When the adversary terminates, all the recorded verification queries are evaluated with respect to the block cipher used in the hybrid at that point, and win is set to 1 if at least one of them is accepted. We denote the resulting augmented hybrids by $\widehat{\mathbf{H}}^z_j$. (If the adversary terminates before the switching point of a hybrid is reached, we sample $\mathbf{K} = (K_1, \ldots, K_u)$ uniformly at random and use the reprogrammed block cipher $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$ only for the final evaluation of the recorded verification queries.)

By construction, $\widehat{\mathbf{H}}^0_1$ is identical to $\mathbf{H}_3$ in the current proof. Indeed, at $z = 0$ the reprogrammed cipher is just E, so the encryption oracle is the real encryption oracle and the final verification is performed with respect to E. On the other hand, since at most m response blocks are produced by the encryption oracle, the

switching point of $\widehat{\mathbf{H}}_1^m$ is never reached. Hence, the encryption oracle remains the ideal encryption oracle \$ throughout the execution, and at termination, the verification queries are evaluated with respect to $E[\![\mathsf{T}_{\mathsf{all}}^{\mathsf{enc}}, \mathbf{K}]\!]$. Thus, $\widehat{\mathbf{H}}_1^m$ is identical to $\mathbf{H}_4$ in the current proof.

We next observe that all the hybrid transitions in the proof of Theorem 1 continue to hold for the augmented hybrids with exactly the same bounds.

First, consider the transition from $\widehat{\mathbf{H}}_1^z$ to $\widehat{\mathbf{H}}_2^z$, corresponding to Lemma 5. We use exactly the same distinguisher $\mathcal{A}'$ for the resampling lemma (Lemma 2) as in the proof of Lemma 5, except that, after the adversary $\mathcal{A}$ terminates, the distinguisher $\mathcal{A}'$ evaluates the recorded verification queries with respect to the block cipher of the corresponding hybrid and outputs the resulting value of win. These additional evaluations occur in the second stage of the experiment of the resampling lemma. Since the bound of Lemma 2 depends only on the number of quantum queries made in its first stage, the additional evaluations do not change the bound. Consequently, the difference between the probabilities of $\widehat{\mathbf{H}}_1^z$ to $\widehat{\mathbf{H}}_2^z$ is upper bounded in the same way as in Lemma 5.

Next, consider the transitions from $\widehat{\mathbf{H}}_2^z$ to $\widehat{\mathbf{H}}_3^z$ and from $\widehat{\mathbf{H}}_3^z$ to $\widehat{\mathbf{H}}_4^z$, corresponding to Lemmas 6 and 7. The bad event used in those lemmas is unchanged, because the verification queries do not affect the adversary's view and do not affect the encryption transcript. Conditioned on $\neg\mathsf{bad}$, the corresponding block-cipher oracles in the two hybrids are identical. Moreover, the recorded verification transcripts are also identical, since the verification oracle always returns $\perp$. Hence, the final verification outcomes are identical as well. It follows that the proofs of Lemmas 6 and 7 apply without any change, with exactly the same bounds.

Finally, consider the transition from $\widehat{\mathbf{H}}_4^z$ to $\widehat{\mathbf{H}}_1^{z+1}$, corresponding to Lemma 8. We modify the distinguisher $\mathcal{A}'$ used in the proof of Lemma 8 only in its final step. After the randomness used to define F_1, in particular the keys $\mathbf{K}$, is revealed in the third stage of the experiment in the arbitrary reprogramming lemma (Lemma 1), the distinguisher $\mathcal{A}'$ continues the simulation exactly as in the proof of Lemma 8. When the adversary $\mathcal{A}$ terminates, the distinguisher $\mathcal{A}'$ evaluates all recorded verification queries with respect to the block cipher of the corresponding hybrid and outputs the resulting value of win. Recall that the distinguisher in the proof of Lemma 8 samples E itself. Therefore, once $\mathbf{K}$ is revealed, this final verification can be performed in the third stage of the experiment for $\mathcal{A}'$ without making any additional query to the oracle given in the experiment of Lemma 1. In particular, the quantity p_z appearing in Lemma 8 is unchanged. The event mcoll and its probability bound are also unchanged, since they depend only on the encryption transcript. Thus, the bound of Lemma 8 continues to hold for the augmented hybrids $\widehat{\mathbf{H}}_4^z$ and $\widehat{\mathbf{H}}_1^{z+1}$.

Therefore, every transition of the augmented hybrids $\widehat{\mathbf{H}}_j^z$ here has exactly the same upper bound as the corresponding transition in the proof of Theorem 1. Applying the same argument for the proof of Theorem 1 around Eq. (27), we

obtain

$$\begin{aligned}&|\Pr[\mathsf{win}=1 \text{ in } \mathbf{H}_3] - \Pr[\mathsf{win}=1 \text{ in } \mathbf{H}_4]| \\ &\quad = \left|\Pr[\mathsf{win}=1 \text{ in } \widehat{\mathbf{H}}_1^0] - \Pr[\mathsf{win}=1 \text{ in } \widehat{\mathbf{H}}_1^m]\right| \le \delta_{\mathsf{priv}}.\end{aligned}$$

□

Lemma 11. $\Pr[\mathsf{win}=1$ *in* $\mathbf{H}_4]$ *is upper bounded by*

$$\begin{aligned}&q_d\left(\frac{nc\ell_{\max}^H}{2^\tau} + \frac{\left(\ell_{\max}^H n+1\right)q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{\ell_{\max}^H n+2}{2^\tau}\right) \\ &\quad + (\sigma_e+q_e+q_d)\left(\frac{(n\ell_{\max}^H+4)Q_e + (n+5)(B_e+1)}{2^n} + \frac{2d}{2^k}\right) \\ &\quad + \ell_{\max}^H 2^{-(\ell_{\max}^H-2)n/2+3} + \frac{2}{2^{2n}} + \frac{(\sigma_e+q_e)^{d+1}}{(d+1)!2^{(d-1)n}}. \end{aligned} \tag{33}$$

Proof. Since $\mathbf{K}$ is independent of the execution of $\mathcal{A}$ in $\mathbf{H}_4$ and is not used until $\mathcal{A}$ stops, we may equivalently sample $\mathbf{K}$ at the beginning for the purpose of bounding the probability of some events.

Let $\Lambda_1 := \left\lceil \frac{\ell_{\max}^H n}{2} 2^{k-n} \right\rceil$ and $\Lambda_2 := \lceil n2^{k-n} \rceil$ as in the proof of Lemma 6, load_1 and load_2 be the events from the proof of Lemma 6, and mcoll be the event from the proof of Lemma 8. The proof of Lemma 6 shows

$$\Pr[\mathsf{load}_1] \le \ell_{\max}^H \cdot 2^{-(\ell_{\max}^H-2)n/2+3}, \qquad \Pr[\mathsf{load}_2] \le 2^{-2n}, \tag{34}$$

and, exactly as in the proof of Lemma 8,

$$\Pr[\mathsf{mcoll}] \le 2^n \binom{\sigma_e+q_e}{d+1} 2^{-dn} \le \frac{(\sigma_e+q_e)^{d+1}}{(d+1)!2^{(d-1)n}}. \tag{35}$$

We also define load_0 to be the event that

$$\left|\{K \in \{0,1\}^k : E_K(0) = L\}\right| > n2^{k-n}$$

for some $L \in \{0,1\}^n$. By applying Lemma 3 with $q = 2^k$, $a = 2^{k-n}$, and $\ell = 2$, we obtain

$$\Pr[\mathsf{load}_0] \le 2^{-3n}. \tag{36}$$

Next, we isolate the event on which all the reprogramming operations used to define $E[\![\mathsf{T}_{\mathsf{all}}^{\mathsf{enc}}, \mathbf{K}]\!]$ from E are compatible with each other. For each $j \in \{1, \ldots, \sigma_e + q_e\}$, let $\mathrm{id}(j)$ be the key index associated with the jth response block in the encryption queries (counted from the beginning), and define x_j, W_j, and $\widetilde{x}_j$ as in the proof of Lemma 6. The reprogramming associated with the jth response block (see Eq. (4) and Eq. (5) for details) swaps the two points

$$\left(K_{\mathrm{id}(j)}, x_j\right) \quad \text{and} \quad \left(K_{\mathrm{id}(j)}, \widetilde{x}_j\right),$$

where

$$\widetilde{x}_j = E_{K_{\mathrm{id}(j)}}^{-1}\left(W_j\left(E_{K_{\mathrm{id}(j)}}(0)\right)\right).$$

Define $\mathsf{bad}_{\mathsf{repro}}$ to be the event that at least one of the following conditions holds:

(i) there exist $1 \leq h < j \leq \sigma_e + q_e$ such that $K_{\mathrm{id}(h)} = K_{\mathrm{id}(j)}$ and at least one of the following equalities holds:

$$x_j = x_h, \qquad x_j = \widetilde{x}_h, \qquad \widetilde{x}_j = x_h, \quad \text{or} \quad \widetilde{x}_j = \widetilde{x}_h.$$

(ii) $\widetilde{x}_j = 0$ for some $j \in \{1, \ldots, \sigma_e + q_e\}$.

Notice that $x_j \neq 0$ for every j, by the definition of the inputs used in CAU.

Below, we bound the probability $\Pr\left[\mathsf{bad}_{\mathsf{repro}} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll}\right]$. Fix $j \in \{1, \ldots, \sigma_e + q_e\}$, and suppose that neither condition (i) nor condition (ii) has occurred for any of the first $(j-1)$ response blocks. Conditioned additionally on $\neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll}$, the event

$$K_{\mathrm{id}(j)} = K_{\mathrm{id}(h)} \quad \text{and} \quad x_j \in \{x_h, \widetilde{x}_h\}$$

for some $h < j$ correspond to Conditions 1 and 2 in the definition of bad in the proof of Lemma 6 (p. 17). The analysis in Eq. (7) and (13) of that proof therefore give

$$\begin{aligned} &\Pr\left[\begin{array}{c} K_{\mathrm{id}(j)} = K_{\mathrm{id}(h)} \text{ and } x_j \in \{x_h, \widetilde{x}_h\} \text{for some } h < j \wedge \\ \neg\mathsf{bad}_{\mathsf{repro}} \text{ until the } j\text{th block} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll} \end{array}\right] \\ &\quad \leq \frac{Q_e\Lambda_1 + B_e\Lambda_2 + 2d}{2^k} \leq \frac{n\ell^H_{\max}Q_e + (n+1)B_e}{2^n} + \frac{2d}{2^k}. \end{aligned} \tag{37}$$

Similarly, the event

$$K_{\mathrm{id}(j)} = K_{\mathrm{id}(h)} \quad \text{and} \quad \widetilde{x}_j \in \{x_h, \widetilde{x}_h\}$$

for some $h < j$, together with the event $\widetilde{x}_j = 0$, correspond to Conditions 3, 4, and 6 in the definition of bad in the proof of Lemma 6. The analysis of Eq. (14), (15), and (17) of that proof give

$$\begin{aligned} &\Pr\left[\begin{array}{c} K_{\mathrm{id}(j)} = K_{\mathrm{id}(h)} \text{ and } \tilde{x}_j \in \{x_h, \widetilde{x}_h\} \text{ for some } h < j \ \vee (\widetilde{x}_j = 0), \text{ and} \\ \neg\mathsf{bad}_{\mathsf{repro}} \text{ until the } j\text{th block} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll} \end{array}\right] \\ &\quad \leq \frac{4(Q_e + B_e) + 1}{2^n}. \end{aligned} \tag{38}$$

From Eq. (37) and (38),

$$\begin{aligned} &\Pr\left[\begin{array}{c} \mathsf{bad}_{\mathsf{repro}} \text{ occurs for the first time at the } j\text{th block} \wedge \\ \neg\mathsf{bad}_{\mathsf{repro}} \text{ until the } j\text{th block} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll} \end{array}\right] \\ &\quad \leq \frac{(n\ell^H_{\max} + 4)Q_e + (n+5)B_e + 1}{2^n} + \frac{2d}{2^k} \end{aligned} \tag{39}$$

follows. Taking a union bound over $j = 1, \ldots, \sigma_e + q_e$, we obtain

$$\begin{aligned} &\Pr\left[\mathsf{bad}_{\mathsf{repro}} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll}\right] \\ &\quad \leq (\sigma_e + q_e)\left(\frac{(n\ell^H_{\max} + 4)Q_e + (n+5)B_e + 1}{2^n} + \frac{2d}{2^k}\right) \end{aligned} \tag{40}$$

Next, let good be the event defined as

$$\mathsf{good} := \neg\mathsf{bad}_{\mathsf{repro}} \wedge \neg\mathsf{load}_0 \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll}.$$

Then

$$\Pr[\neg\mathsf{good}] \leq (\sigma_e + q_e)\left(\frac{(n\ell^H_{\max}+4)Q_e + (n+5)B_e + 1}{2^n} + \frac{2d}{2^k}\right) + \ell^H_{\max} 2^{-(\ell^H_{\max}-2)n/2+3} + \frac{2}{2^{2n}} + \frac{(\sigma_e+q_e)^{d+1}}{(d+1)!2^{(d-1)n}}. \tag{41}$$

follows from Eq. (34)-(36) and (40).

In what follows, we will bound the probability of $\mathsf{win} = 1$ conditioned on good. Conditioned on good, all the values swapped in defining $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$ from E are pairwise disjoint, every programmed encryption input has its prescribed output. In addition, every point that does not appear in the definition of $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$ (Eq. (4) and (5)) retains its value under E. In particular,

$$E[\![\mathsf{T}^{\mathrm{enc}}_{\mathrm{all}}, \mathbf{K}]\!]_{K_i}(0) = E_{K_i}(0) \tag{42}$$

for every i.

Fix a verification query $V := (i, N, AD, \tilde{C}, \tilde{T})$, and write $Y := \mathsf{pad}(N)$, $X := (AD, \tilde{C})$, and $L := E_{K_i}(0)$. Below, we bound the probability that $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}(V) = \top$. Let $\mathsf{win}_1(V)$ and $\mathsf{win}_2(V)$ be the events defined as follows.

$\underline{\mathsf{win}_1(V)}$: $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}(V) = \top$ holds, and $\mathsf{T}^{\mathrm{enc}}_{\mathrm{all}}$ contains an encryption query of the form (i, N, AD', M') for some AD' and M' (the response is denoted by (C', T')).
$\underline{\mathsf{win}_2(V)}$: $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}(V) = \top$ holds, and $\mathsf{T}^{\mathrm{enc}}_{\mathrm{all}}$ does *not* contain an encryption query of the form (i, N, AD', M') for any AD' and M'.

Recall that E and the complete encryption and verification transcripts are determined before $\mathbf{K}$ is sampled in $\mathbf{H}_4$.

First, we bound the probability of $\mathsf{win}_1(V)$, conditioned on good. The encryption query of the form (i, N, AD', M') is unique by nonce-respectingness. Put $X' := (AD', \tilde{C}')$ (here, $\tilde{C}' := \mathsf{msb}_{|M'|}(C')$). Conditioned on good, acceptance of the verification query implies

$$H_L(X) \oplus H_L(X') = \left(\tilde{T} || Z\right) \oplus T' \tag{43}$$

for some $Z \in \{0,1\}^{n-\tau}$. If $X \neq X'$, then since H is c-AXU, the set of L satisfying Eq. (43) has size at most[9] $c\ell^H_{\max}$ (per each Z). Conditioned on $\neg\mathsf{load}_0$, each such L has at most $n2^{k-n}$ preimage keys $K \in \{0,1\}^k$ under the mapping $K \mapsto E_K(0)$, and the number of possible Z is $2^{n-\tau}$. Hence

$$\Pr[\mathsf{win}_1(V) \wedge X \neq X' \wedge \mathsf{good}] \leq \frac{c\ell^H_{\max} \cdot n2^{k-n} \cdot 2^{n-\tau}}{2^k} = \frac{nc\ell^H_{\max}}{2^\tau}.$$

[9] The argument of bounding the number of L satisfying the equation is based on the corresponding arguments using balls-into-bins in [HTT18].

If $X = X'$, acceptance requires $\tilde{T} = \mathsf{msb}_\tau(T')$. When the encryption response precedes the verification query, this is a prohibited replay. When the encryption query occurs later, T' is freshly and uniformly sampled by the ideal encryption oracle after the verification query has been fixed. Therefore,

$$\Pr[\mathsf{win}_1(V) \wedge X = X' \wedge \mathsf{good}] \leq 2^{-\tau}.$$

In summary, we have

$$\Pr[\mathsf{win}_1(V) \wedge \mathsf{good}] \leq \frac{nc\ell_{\max}^H}{2^\tau} + 2^{-\tau}. \tag{44}$$

Second, we bound the probability of $\mathsf{win}_2(V)$ conditioned on good. We introduce additional events kcoll, $\mathsf{coll}^1_{\mathsf{repro}}$, and $\mathsf{coll}^2_{\mathsf{repro}}$ as follows.

1. kcoll: $\mathsf{win}_2(V)$ happens, and there is an encryption query (i', N', AD', M') recorded in $\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}$ such that $i' \neq i$, $N' = N$, and $K_i = K_{i'}$.
2. $\mathsf{coll}^1_{\mathsf{repro}}$: $\mathsf{win}_2(V)$ happens, there is $j \leq q_e + \sigma_e$ such that $Y(= \mathsf{pad}(N)) = \tilde{x}_j$, and the jth block is in the response to an encryption query with the ith key.
3. $\mathsf{coll}^2_{\mathsf{repro}}$: $\mathsf{win}_2(V)$ happens, there is $j \leq q_e + \sigma_e$ such that $Y(= \mathsf{pad}(N)) = \tilde{x}_j$, the jth block is in the response to an encryption query with the i'th key for some $i' \neq i$, and $K_{i'} = K_i$.

We will bound $\Pr[\mathsf{kcoll} \wedge \mathsf{good}]$. If $\mathsf{win}_2(V)$ happens, then there is no encryption query with the pair (i, N). There are at most d encryption queries with nonce N (with the key index other than i). Hence,

$$\Pr\left[\mathsf{kcoll} \wedge \mathsf{good}\right] \leq \frac{d}{2^k}. \tag{45}$$

Next, we will bound $\Pr\left[\mathsf{coll}^1_{\mathsf{repro}} \wedge \mathsf{good}\right]$. For a non-tag response block of the ith key, the equality $Y = \widetilde{x}_j$ is equivalent to

$$E_{K_i}(Y) = W_j,$$

and for a tag block it is equivalent to

$$E_{K_i}(Y) \oplus H_{E_{K_i}(0)}(AD_j, \tilde{C}_j) = T_j.$$

On $\neg\mathsf{load}_2$ and $\neg\mathsf{load}_1$, respectively, these equations have at most Λ_2 and Λ_1 solutions[10] for K_i. There are at most B_e non-tag blocks and Q_e tag blocks for the ith key. Hence,

$$\Pr\left[\mathsf{coll}^1_{\mathsf{repro}} \wedge \mathsf{good}\right] \leq \frac{B_e\Lambda_2 + Q_e\Lambda_1}{2^k} \leq \frac{\left(\ell_{\max}^H n + 1\right) Q_e + (n+1)B_e}{2^n}. \tag{46}$$

Similarly,

$$\Pr\left[\mathsf{coll}^2_{\mathsf{repro}} \wedge \mathsf{good}\right] \leq \frac{\sigma_e\Lambda_2 + q_e\Lambda_1}{2^{2k}} \leq \frac{\left(\ell_{\max}^H n + 1\right) q_e + (n+1)\sigma_e}{2^{n+k}}. \tag{47}$$

[10] The argument of bounding the number of solutions here is also based on the corresponding balls-into-bins arguments in [HTT18].

holds (note that the condition $K_{i'} = K_i$ is imposed here, and that we have σ_e and q_e instead of B_e and Q_e because $\mathsf{coll}^2_{\mathsf{repro}}$ concerns the key indices i' that are not i).

Conditioned on $\neg\mathsf{kcoll}$, $\neg\mathsf{coll}^1_{\mathsf{repro}}$, and $\neg\mathsf{coll}^2_{\mathsf{repro}}$ the value (K_i, Y) does not appear in defining $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$ from E as in Eq. (4) and (5), and thus $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!](Y) = E_{K_i}(Y)$. In addition, conditioned on good, $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!](0) = E_{K_i}(0)$ holds by Eq. (42). Thus, $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}(V) = \mathsf{Vrf}^{E}(V)$ holds for the verification query $V = (i, N, AD, \tilde{C}, \tilde{T})$, and the response is $\top$ only if

$$E_{K_i}(Y) \oplus H_{E_{K_i}(0)}(X) = \tilde{T}||Z \tag{48}$$

for some $Z \in \{0,1\}^{n-\tau}$. Since Eq. (48) has at most Λ_1 solutions for K_i under good (especially $\neg\mathsf{load}_1$), by taking the union bound on Z, we have

$$\begin{aligned}\Pr\left[\mathsf{win}_2(V) \wedge \mathsf{good} \wedge \neg\mathsf{kcoll} \wedge \neg\mathsf{coll}^1_{\mathsf{repro}} \wedge \neg\mathsf{coll}^2_{\mathsf{repro}}\right] &\leq \frac{\Lambda_1}{2^k} \cdot 2^{n-\tau} \\ &\leq \frac{(\ell^H_{\max} n + 1)}{2^\tau}. \end{aligned} \tag{49}$$

Combining Eq. (45)-(47) and (49), we obtain

$$\begin{aligned}\Pr\left[\mathsf{win}_2(V) \wedge \mathsf{good}\right] \leq{}& \frac{\left(\ell^H_{\max} n + 1\right) Q_e + (n+1)(B_e+1)}{2^n} \\ &+ \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{d}{2^k} \\ &+ \frac{\ell^H_{\max} n + 1}{2^\tau}. \end{aligned} \tag{50}$$

By taking the union bound for all the verification queries,

$$\begin{aligned}&\Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_4 \wedge \mathsf{good}] \\ &\leq \Pr\left[\begin{array}{l}\mathsf{win}_1(V) \text{ for some} \\ \text{verification query } V\end{array} \wedge \mathsf{good}\right] + \Pr\left[\begin{array}{l}\mathsf{win}_2(V) \text{ for some} \\ \text{verification query } V\end{array} \wedge \mathsf{good}\right] \\ &\leq q_d \left(\frac{nc\ell^H_{\max}}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) Q_e + (n+1)(B_e+1)}{2^n} \right. \\ &\qquad \left. + \frac{\ell^H_{\max} n + 2}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{d}{2^k} \right) \end{aligned} \tag{51}$$

follows from Eq. (44) and (50).

Consequently,

$$
\begin{aligned}
&\Pr[\mathsf{win}=1 \text{ in } \mathbf{H}_4] \\
&\le \Pr[\mathsf{win}=1 \text{ in } \mathbf{H}_4 \wedge \mathsf{good}] + \Pr[\neg\mathsf{good}] \\
&\le q_d \left(\frac{nc\ell^H_{\max}}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) Q_e + (n+1)(B_e+1)}{2^n} \right. \\
&\qquad \left. + \frac{\ell^H_{\max} n + 2}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{d}{2^k} \right) \\
&\quad + (\sigma_e + q_e) \left(\frac{(n\ell^H_{\max}+4)Q_e + (n+5)B_e + 1}{2^n} + \frac{2d}{2^k} \right) \\
&\quad + \ell^H_{\max} 2^{-(\ell^H_{\max}-2)n/2+3} + \frac{2}{2^{2n}} + \frac{(\sigma_e+q_e)^{d+1}}{(d+1)!2^{(d-1)n}} \\
&\le q_d \left(\frac{nc\ell^H_{\max}}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{\ell^H_{\max} n + 2}{2^\tau} \right) \\
&\quad + (\sigma_e + q_e + q_d) \left(\frac{(n\ell^H_{\max}+4)Q_e + (n+5)(B_e+1)}{2^n} + \frac{2d}{2^k} \right) \\
&\quad + \ell^H_{\max} 2^{-(\ell^H_{\max}-2)n/2+3} + \frac{2}{2^{2n}} + \frac{(\sigma_e+q_e)^{d+1}}{(d+1)!2^{(d-1)n}} \qquad (52)
\end{aligned}
$$

follows from Eq. (41). □

Finishing the Proof of Theorem 2. Since $\Pr[\mathsf{win}=1 \text{ in } \mathbf{H}_1]$ upper bounds the probability of an adversary forging, the claim of the theorem follows from Lemma 9, 10, and 11.

6 Security when Nonces are Randomized

As shown in the previous works in the classical setting [BT16,HTT18], the multi-key security bounds of GCM (CAU) can be improved when the nonce-generation mechanism is randomized. Hoang et al. [HTT18] study nonce-randomization techniques based on XOR and concatenation, mainly treating the randomizer as part of the secret key, and also discuss public salting for the XOR-based construction. Motivated by their work, we consider the following two public-salt variants.

1. (Adding a random salt to per-key distinct offsets.) A random public r-bit salt $R_i \in \{0,1\}^r$ is chosen (for each key index i), and the honest parties use nonces $R_i + c_{i,0}, R_i + c_{i,1}, R_i + c_{i,2}, \ldots$ in a sequential order (here, "+" is an arbitrary abelian group operation defined over $\{0,1\}^r$), where $c_{i,j}$ are pre-defined constants such that $c_{i,j} \neq c_{i,j'}$ for $j \neq j'$. The typical case is that these constants represent counter or sequence-number values.

2. (Concatenating a random salt.) A parameter $t \leq r$ is chosen, a random t-bit public salt $R'_i \in \{0,1\}^t$ is chosen (for each key index i), and nonces are always of the form $R'||N$ for some $N \in \{0,1\}^{r-t}$.

The first (resp., the second) variant can be regarded as the nonce-generation patterns of AES-GCM used in TLS 1.3 [Res26], QUIC [TT21], SRTP [MI15] (resp., TLS 1.2 [SCM08] and IPsec ESP [VM05]), and the successor-based instantiation of the RBG-based construction of NIST SP 800-38D [Dwo07]. In the first type, a fixed per-key nonce component is combined with a counter- or sequence-dependent value by XOR (or, more generally, a group operation), whereas in the second type it is concatenated with an explicit per-record or per-packet value.

In several of these protocols, the fixed per-key component corresponding to our random salt is derived from secret keying material using a KDF, rather than sampled as an independent public random string. We abstract away this key-derivation layer and model the fixed component as an independently uniform public salt. If an independently sampled salt is kept secret rather than revealed, our public-salt security bound continues to hold, since revealing the salt only gives additional information to the adversary. (Strictly speaking, relating KDF-derived fixed components to this idealized model additionally requires an appropriate post-quantum pseudorandomness assumption on the KDF, but we do not get into more details about this.)

We show that these two nonce-generation mechanisms yield improved post-quantum multi-key security bounds for GCM (CAU). The reasoning is essentially the same as the one discussed in the classical setting [HTT18], but this section provides details for completeness.

First, regarding the first modification of adding a public random salt to per-key distinct offsets, the following corollary holds.

Corollary 1 (Security of CAU, adding a public random salt to per-key distinct offsets). *Suppose that all the assumptions of Theorem 1 (resp., Theorem 2) hold, except for the assumption that every nonce is queried for at most d keys. Assume additionally that the jth classical encryption query by $\mathcal{A}$ associated with the ith key is of the form $(i, R_i + c_{i,j-1}, AD, M)$ for some AD and M, where $R_1, \ldots, R_u \in \{0,1\}^r$ are (public) uniform and independent random salts sampled at the beginning of the game, $u \cdot Q_e < 2^r$, and $c_{i,j}$ are pre-defined constants such that $c_{i,j} \neq c_{i,j'}$ for $j \neq j'$. Then, $\mathcal{A}$'s advantage is upper bounded as $\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + 2^{-r/2}$ (resp., $\mathsf{Adv}^{\mathsf{mk\text{-}auth}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \delta' + 2^{-r/2}$), where δ_{priv} is as in Theorem 1 (δ_{priv} and δ' are as in Theorem 2) and the parameter d in δ_{priv} (d in $\delta_{\mathsf{priv}} + \delta'$) can be arbitrarily set to be any positive integer such that $d \geq \lceil 1.5/(1 - (\log_2(uQ_e)/r)) \rceil - 1$.*

Proof. We give a proof[11] for $\mathsf{mk\text{-}priv}$ (the proof for $\mathsf{mk\text{-}auth}$ is analogous). Let bad be an event that $\mathcal{A}$ queries a single nonce to the encryption oracle for more than d keys. Then, $\Pr[\mathsf{bad}] \leq 2^{-r/2}$ follows from Lemma 4, by setting the variables X^i_j as $X^i_j := R_i + c_{i,j-1}$ (here, we assume that the variables X^i_j are defined before $\mathcal{A}$

[11] This proof is written with the assistance of ChatGPT.

starts). As long as bad does not happen, the setting is the same as Theorem 1, so we have $\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \Pr[\mathsf{bad}] \leq \delta_{\mathsf{priv}} + 2^{-r/2}$. □

Second, regarding the second modification of concatenating a public random salt, the following corollary holds.

Corollary 2 (Security of CAU with concatenating a public random salt). *Let t and d be positive integers such that $t \leq r$. Suppose that all the assumptions of Theorem 1 (resp., Theorem 2) hold, except for the assumption that every nonce is queried for at most d keys. Assume additionally that the nonces queried to the encryption oracle are always of the form $R'_i||N'$ for some $N' \in \{0,1\}^{r-t}$, where $R'_1, \ldots, R'_u \in \{0,1\}^t$ are (public) uniform and independent random salts sampled at the beginning of the game. Then, $\mathcal{A}$'s advantage is upper bounded as $\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + (u^{d+1}/(d+1)!2^{dt})$ (resp., $\mathsf{Adv}^{\mathsf{mk\text{-}auth}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \delta' + (u^{d+1}/(d+1)!2^{dt})$), where δ_{priv} is as in Theorem 1 (δ_{priv} and δ' are as in Theorem 2).*

Proof. We give a proof for mk-priv (the proof for mk-auth is analogous). Let bad be an event that $\mathcal{A}$ queries a single nonce to the encryption oracle for more than d keys. Then, bad happens only when the values of $(d+1)$ random t-bit salts collide, and thus $\Pr[\mathsf{bad}] \leq \binom{u}{d+1}/2^{dt} \leq (u^{d+1}/(d+1)!2^{dt})$. As long as bad does not happen, the setting is the same as Theorem 1, so we have $\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \Pr[\mathsf{bad}] \leq \delta_{\mathsf{priv}} + (u^{d+1}/(d+1)!2^{dt})$. □

7 Interpretation of the Security Bounds

This section discusses interpretation of the security bounds shown in Theorem 1 and Theorem 2 (and the corollaries in the previous section).

Throughout this section, we use the quantum-query complexity at which the relevant security bound reaches constant advantage as a coarse measure of security. More precisely, after fixing the classical-query parameters, we compare the value of p at which the p-dependent part of the bound becomes of constant order. This provides only a coarse comparison of the bounds in terms of query complexity, and should not be interpreted as a usage limit for a prescribed small target advantage. For a fixed target advantage such as 2^{-32} or 2^{-60}, the quantitative comparison may be different. Moreover, since we compare upper bounds, reaching constant order only means that the bound ceases to guarantee small advantage; it does not imply a matching attack with constant success probability.

Overall, our bounds say that the conclusion analogous to the classical work by Hoang et al. [HTT18] also holds in the post-quantum setting; we can achieve better multi-key security bounds compared to the trivial ones obtained by multiplying the number of keys u to the corresponding single-key bound. In particular, the u-dependent quantum term $up^2/2^k$ arising from the trivial multi-key reduction is replaced by a d-dependent term of order $\sqrt{dp^2/2^k}$, at the cost of an additional term of order $\sqrt{\sigma_e^2 p/2^k}$.

In what follows, we provide more detailed discussions on privacy and authenticity separately, assuming that CAU is instantiated with AES ($n = 128$) and the nonce length r is set as $r = 96$, and nonces are generated by adding a random salt with per-key distinct offsets (e.g., counter values) as in Corollary 1. According to Corollary 1, we can use arbitrary d satisfying $d \geq \lceil 1.5/(1 - (\log_2(uQ_e)/r)) \rceil - 1$ (an additional term $2^{-r/2}$ appears in the bound, but this does not play an important role in considering the query complexity for the advantages to reach a constant order, and we omit it from this coarse comparison). We will conservatively set $Q_e = 2^{32}$, which matches the 2^{32}-invocation-per-key limit prescribed by NIST SP 800-38D for the RBG-based IV construction [Dwo07] and is substantially larger than the current TLS 1.3 AES-GCM limit of $2^{24.5}$ full-size records under a given set of keys [Res26].

On Privacy. First, we study the interpretation of the privacy bound (Theorem 1 and Corollary 1).

If $\ell^H_{\max} \geq 4$ and $d \geq 5$, then the terms $\frac{16(\sigma_e+q_e)\ell^H_{\max}}{2^{(\ell^H_{\max}-2)n/2}}$ and $\frac{4(\sigma_e+q_e+1)^{d+2}}{(d+1)!\,2^{(d-1)n}}$ in Eq. (2) become negligible compared to other terms. Assuming that $\sigma_e \gg q_e$ and that d is a small constant, the dominant terms in the bound of Theorem 1 are

$$\sqrt{\frac{16\sigma_e^2 p}{2^k}} + \sqrt{\frac{16p^2 d}{2^k}} + \frac{2n(\ell^H_{\max} Q_e + B_e)\sigma_e + 4\sigma_e^2}{2^n} + \frac{2\sigma_e B_e}{2^k}. \tag{53}$$

The third and fourth terms of Eq. (53) do not involve p and depend only on the classical encryption queries. They are significantly smaller than 1 as long as $\sigma_e \ll 2^{n/2-1}$, $B_e\sigma_e \ll 2^{n-1-\log_2 n}$, and $\ell^H_{\max} Q_e \sigma_e \ll 2^{n-1-\log_2 n}$

Meanwhile, the first two terms of Eq. (53) depend on the parameter p, which is the number of quantum queries to the cipher E and the inverse E^{-1}. The second term $\sqrt{\frac{16p^2 d}{2^k}}$ is significantly smaller than 1 as long as $p \ll 2^{k/2-2-\frac{1}{2}\log_2 d}$. The tricky term is the first one, which is a kind of an artifact of the proof technique, as we will detail later.

To analyze the effect of the first term $\sqrt{\frac{16\sigma_e^2 p}{2^k}}$, let us estimate σ_e as $\sigma_e \approx uB_{\mathsf{mean}}$, where B_{mean} is the average of the number of the blocks processed by the encryption function per key. The value B_{mean} heavily depends on applications and could be large sometimes, but remains small in some practical protocols. For example, Cloudflare [AW25] reports mean server-to-client TCP payload volumes of 224 KB and 390 KB per HTTP/1.x and HTTP/2 connection, respectively. Dividing these values by the 16-byte AES block size gives workload proxies of approximately $2^{13.8}$ and $2^{14.6}$ blocks per connection, suggesting that $B_{\mathsf{mean}} = 2^{15}$ is a reasonable illustrative workload value. If $B_{\mathsf{mean}} = 2^{15}$, then the first term $\sqrt{\frac{16\sigma_e^2 p}{2^k}} \approx \sqrt{\frac{16u^2 B^2_{\mathsf{mean}} p}{2^k}}$ is significantly smaller than 1 as long as $p \ll 2^{k/2-2-\frac{1}{2}\log_2 d}$ when $u \leq 2^{k/4-16+\frac{1}{4}\log_2 d}$.

To illustrate the significance of the bound, assume $k = 192$, and set $u = 2^{k/4-16} = 2^{32}$. In this case, $d = 5$ suffices to apply Corollary 1 (because we are assuming $Q_e = 2^{32}$), and thus the first two terms of Eq. (53) are significantly

smaller than 1 if $p \ll 2^{k/2-2-\frac{1}{2}\log_2 d} \approx 2^{92.8}$. This is in contrast to the trivial multi-key bound obtained from the single-key bound having the term $up^2/2^k$, which does not guarantee security for $p \geq 2^{80}$ when $k = 192$ and $u = 2^{32}$.

On Authenticity. Next, we consider the interpretation of the authenticity bounds (Theorem 2).

In Theorem 2, the bound is given as $\delta_{\mathsf{priv}} + \delta'$. Assuming $\ell^H_{\max} \geq 4$ and $d \geq 5$, the terms $\frac{16(\sigma_e+q_e)\ell^H_{\max}}{2^{(\ell^H_{\max}-2)n/2}} + \frac{4(\sigma_e+q_e+1)^{d+2}}{(d+1)!\,2^{(d-1)n}}$ (in δ_{priv}) and $\ell^H_{\max}2^{-(\ell^H_{\max}-2)n/2+3} + \frac{(\sigma_e+q_e)^{d+1}}{(d+1)!2^{(d-1)n}}$ (in δ') become negligible compared to other terms. Additionally assuming that $\sigma_e \gg q_e$ and that d is a small constant, the dominant terms are approximately equal to

$$\begin{aligned}
&\sqrt{\frac{16\sigma_e^2 p}{2^k}} + \sqrt{\frac{16p^2 d}{2^k}} + \frac{2n(\ell^H_{\max}Q_e + B_e)\sigma_e + 4\sigma_e^2}{2^n} + \frac{2\sigma_e B_e}{2^k} \\
&\quad + q_d\left(\frac{n(c+1)\ell^H_{\max}}{2^\tau} + \frac{\ell^H_{\max}nq_e + n\sigma_e}{2^{n+k}}\right) + (\sigma_e + q_d)\left(\frac{n\ell^H_{\max}Q_e + nB_e}{2^n}\right) \\
&\lessapprox \sqrt{\frac{16\sigma_e^2 p}{2^k}} + \sqrt{\frac{16p^2 d}{2^k}} + \frac{3n(\ell^H_{\max}Q_e + B_e)\sigma_e + 4\sigma_e^2}{2^n} + \frac{2\sigma_e B_e}{2^k} \\
&\quad + q_d\left(\frac{n(c+1)\ell^H_{\max}}{2^\tau} + \frac{n\ell^H_{\max}Q_e + nB_e}{2^n}\right) \qquad (54)
\end{aligned}$$

(note that we dropped the terms with denominator 2^{2n} or 2^{n+k} as those terms are also negligible compared to other terms).

In Eq. (54), the first four terms are in the same order as those appear in the privacy bound, which we have already analyzed. The remaining terms contain only the parameters regarding classical queries. Since the parameter c is at most 1.5 in AES-GCM [HTT18], the remaining terms are significantly smaller than 1 as long as $2.5nq_d\ell^H_{\max} \ll 2^\tau$, $n\ell^H_{\max}Q_eq_d \ll 2^n$, and $nB_eq_d \ll 2^n$.

7.1 On Tightness and Room for Improvements of the Bounds

In both the privacy and authenticity bounds (Theorems 1 and 2), the terms that involve p (the number of quantum queries to E and E^{-1}) are those of the order $\sqrt{p^2d/2^k}$ and $\sqrt{\sigma_e^2 p/2^k}$. The former term $\sqrt{p^2d/2^k}$ roughly reflects the complexity of the multi-target key search by Grover's algorithm (if a single nonce N is queried to the encryption oracle for d keys, then the d-target preimage search with Grover's algorithm recovers one of those d keys with $p \approx \sqrt{2^k/d}$))[12]. Meanwhile, the latter term $\sqrt{\sigma_e^2 p/2^k}$ appears to be an artifact arising from the repeated application of the resampling lemma (Lemma 2). Similar terms also appear in the previous works taking the reprogramming-and-resampling approach [ABK+24,Hos25,BBC+25,SI26], but there are no known

[12] Strictly speaking, the advantage by Grover's algorithm is $p^2d/2^k$, so the term $\sqrt{p^2d/2^k}$ is tight only with respect to the number of queries.

attacks matching those terms. (Due to the latter term, our bound does not guarantee security when σ_e is large. For example, if 2^{32} encryption queries (with non-empty message) are made for every key and $u = 2^{32}$, then the term becomes at least $\sqrt{2^{128}p/2^k}$, and the security is not guaranteed for $p \geq 2^{k-128}$.)

The remaining terms involve only the complexity of classical queries. In the privacy bound, the order of essentially the largest term among them is $O(\sigma_e^2/2^n)$. This roughly reflects the fact that the classical tight bound of GCM (CAU) is the birthday bound, and tight in the single-key setting ($u = 1$). However, the leading term of the classical bound shown in [HTT18] is $O(B_e\sigma_e/2^n)$, and there is room for improvement (if $\sigma_e = uB_e$, then our term is $O(u^2B_e^2/2^n)$ while the classical bound is $O(uB_e^2/2^n)$).

The authenticity bound is the sum of the privacy bound (δ_{priv} of Theorem 1) and the additional term δ' of Eq. (32). In δ', the largest terms among those that we have not discussed are roughly in the order of $q_d\ell_{\max}^H/2^\tau$ and $q_d(\ell_{\max}^H Q_e + B_e)/2^n$. Regarding the first term, the dependence on $q_d/2^\tau$ is tight due to the standard random-tag guessing attack, which accounts for the term $q_d\ell_{\max}^H/2^\tau$ when $\ell_{\max}^H = O(1)$. However, this attack does not explain the dependence on $\ell_{\max}^H$, and we are not aware of an attack exploiting long ciphertexts to obtain such a factor. On the latter term $q_d(\ell_{\max}^H Q_e + B_e)/2^n$, we do not know a matching attack, and this term might be removable with another proof technique.

AI Tool Use Disclosure

OpenAI's ChatGPT (GPT-5.6 Sol) was used as an interactive tool during the preparation of this manuscript. Beyond language editing, it was used at several points to suggest or help refine proof ideas and technical arguments, and to review intermediate arguments for potential gaps or errors. Substantive uses related to the development of proofs are explicitly identified in footnotes at the corresponding locations in the manuscript, where the nature of the assistance is described. The author independently checked and verified all final arguments and takes full responsibility for the correctness of all claims and proofs in this work.

References

ABK+24. Gorjan Alagic, Chen Bai, Jonathan Katz, Christian Majenz, and Patrick Struck. Post-quantum security of tweakable Even-Mansour, and applications. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part I*, volume 14651 of *LNCS*, pages 310–338. Springer, Cham, May 2024.

ABKM22. Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. Post-quantum security of the Even-Mansour cipher. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 458–487. Springer, Cham, May / June 2022.

ABMS26. Gorjan Alagic, Chen Bai, Christian Majenz, and Kaiyan Shi. Post-quantum security of block cipher constructions. Cryptology ePrint Archive, Paper 2026/209, 2026. To appear in Selected Areas in Cryptography (SAC) 2026.

ADL17. Tomer Ashur, Orr Dunkelman, and Atul Luykx. Boosting authenticated encryption robustness with minimal modifications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 3–33. Springer, Cham, August 2017.

Age26. Agence nationale de la sécurité des systèmes d'information (ANSSI), France. Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques. Technical Report ANSSI-PG-083, ANSSI, March 2026. Version 3.00, 20 March 2026.

ATTU16. Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh. Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In Tsuyoshi Takagi, editor, *PQCrypto 2016*, pages 44–63. Springer, Cham, 2016.

AW25. Suleman Ahmad and Peter Wu. Measuring characteristics of TCP connections at internet scale. Cloudflare Blog, October 2025. Published October 29, 2025. Accessed August 4, 2026.

BBC+21. Ritam Bhaumik, Xavier Bonnetain, André Chailloux, Gaëtan Leurent, María Naya-Plasencia, André Schrottenloher, and Yannick Seurin. QCB: Efficient quantum-secure authenticated encryption. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 668–698. Springer, Cham, December 2021.

BBC+25. Jyotirmoy Basak, Ritam Bhaumik, Amit Kumar Chauhan, Ravindra Jejurikar, Ashwin Jha, Anandarup Roy, André Schrottenloher, and Suprita Talnikar. Post-quantum security of key-alternating feistel ciphers. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part I*, volume 16245 of *LNCS*, pages 446–478. Springer, Singapore, December 2025.

BBM00. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Berlin, Heidelberg, May 2000.

BDD26. Ritam Bhaumik, Jean Paul Degabriele, and Chandranan Dhar. ♯Pencil: A Domain-Extended Committing BBB PRF for Strengthening GCM. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology – CRYPTO 2026*, volume 16805 of *Lecture Notes in Computer Science*. Springer, Cham, 2026.

BH22. Mihir Bellare and Viet Tung Hoang. Efficient schemes for committing authenticated encryption. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 845–875. Springer, Cham, May / June 2022.

BHT98. Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In Claudio L. Lucchesi and Arnaldo V. Moura, editors, *LATIN 1998*, volume 1380 of *LNCS*, pages 163–169. Springer, Berlin, Heidelberg, April 1998.

BHT18. Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. Revisiting AES-GCM-SIV: Multi-user security, faster key derivation, and better bounds. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 468–499. Springer, Cham, April / May 2018.

Bih02. Eli Biham. How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. *Information Processing Letters*, 84(3):117–124, 2002.

BN18. Srimanta Bhattacharya and Mridul Nandi. Revisiting variable output length xor pseudorandom function. *IACR Trans. Symm. Cryptol.*, 2018(1):314–335, 2018.

Boz25. Andrey Bozhko. Properties of authenticated encryption with associated data (AEAD) algorithms. RFC 9771, May 2025.

BT16. Mihir Bellare and Björn Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 247–276. Springer, Berlin, Heidelberg, August 2016.

CHK+25. Woohyuk Chung, Seongha Hwang, Seongkwang Kim, Byeonghak Lee, and Jooyoung Lee. Making GCM great again: Toward full security and longer nonces. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part I*, volume 15601 of *LNCS*, pages 33–61. Springer, Cham, May 2025.

CMP24. Matthew Campagna, Alexander Maximov, and John Preuß Mattsson. Galois counter mode with secure short tags (GCM-SST). Internet-Draft draft-mattsson-cfrg-aes-gcm-sst-03, March 2024. Work in Progress.

CR22. John Chan and Phillip Rogaway. On committing authenticated-encryption. In Vijayalakshmi Atluri, Roberto Di Pietro, Christian Damsgaard Jensen, and Weizhi Meng, editors, *ESORICS 2022, Part II*, volume 13555 of *LNCS*, pages 275–294. Springer, Cham, September 2022.

DGRW18. Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast message franking: From invisible salamanders to encryptment. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 155–186. Springer, Cham, August 2018.

Dig26. Digital Agency and Ministry of Internal Affairs and Communications and Ministry of Economy, Trade and Industry, Japan. The list of ciphers that should be referred to in the procurement for the e-government system (CRYPTREC ciphers list). Technical Report CRYPTREC LS-0001-2022R2, CRYPTREC, March 2026. Originally published March 30, 2023; last updated March 30, 2026.

Dwo07. Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. NIST Special Publication 800-38D, National Institute of Standards and Technology, 2007.

EM97. Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–162, June 1997.

FS26. Marc Fischlin and Tobias Schmalz. Quantum-resistant authenticated encryption variants of GCM. *IACR Transactions on Symmetric Cryptology*, 2026(2), 2026.

GL15. Shay Gueron and Yehuda Lindell. GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015*, pages 109–119. ACM Press, October 2015.

GL17. Shay Gueron and Yehuda Lindell. Better bounds for block cipher modes of operation via nonce-based key derivation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1019–1036. ACM Press, October / November 2017.

GLL17. Shay Gueron, Adam Langley, and Yehuda Lindell. AES-GCM-SIV: Specification and analysis. Cryptology ePrint Archive, Report 2017/168, 2017.

GR25. Shay Gueron and Thomas Ristenpart. DNDK: Combining nonce and key derivation for fast and scalable AEAD. Cryptology ePrint Archive, Report 2025/785, 2025.

Gro96. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *28th ACM STOC*, pages 212–219. ACM Press, May 1996.

Hos25. Akinori Hosoyamada. Post-quantum security of keyed sponge-based constructions through a modular approach. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part I*, volume 16245 of *LNCS*, pages 411–445. Springer, Singapore, December 2025.

HTT18. Viet Tung Hoang, Stefano Tessaro, and Aishwarya Thiruvengadam. The multi-user security of GCM, revisited: Tight bounds for nonce randomization. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 1429–1440. ACM Press, October 2018.

HY18. Akinori Hosoyamada and Kan Yasuda. Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 275–304. Springer, Cham, December 2018.

IIM25. Akiko Inoue, Tetsu Iwata, and Kazuhiko Minematsu. Comprehensive robustness analysis of GCM, CCM, and OCB3. In Arpita Patra, editor, *CT-RSA 2025*, volume 15598 of *LNCS*, pages 75–98. Springer, Cham, April / May 2025.

IJMM25. Akiko Inoue, Ashwin Jha, Bart Mennink, and Kazuhiko Minematsu. Generic security of GCM-SST. In Marc Fischlin and Veelasha Moonsamy, editors, *ACNS 2025, Part II*, volume 15826 of *LNCS*, pages 342–369. Springer, Cham, June 2025.

IM16. Tetsu Iwata and Kazuhiko Minematsu. Stronger security variants of GCM-SIV. *IACR Trans. Symm. Cryptol.*, 2016(1):134–157, 2016.

IOM12. Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and repairing GCM security proofs. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 31–49. Springer, Berlin, Heidelberg, August 2012.

IS09. K. Igoe and J. Solinas. AES Galois Counter Mode for the Secure Shell Transport Layer Protocol. RFC 5647, August 2009.

IS17. Tetsu Iwata and Yannick Seurin. Reconsidering the security bound of AES-GCM-SIV. *IACR Trans. Symm. Cryptol.*, 2017(4):240–267, 2017.

ISO20. ISO/IEC. Information security – authenticated encryption. ISO/IEC 19772:2020, November 2020.

Jou06. Antoine Joux. Authentication failures in NIST version of GCM. Public comment on the first draft of NIST Special Publication 800-38D, April 2006.

KLLN16. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 207–237. Springer, Berlin, Heidelberg, August 2016.

KR96. Joe Kilian and Phillip Rogaway. How to protect DES against exhaustive key search. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 252–267. Springer, Berlin, Heidelberg, August 1996.

LL23. Nathalie Lang and Stefan Lucks. On the post-quantum security of classical authenticated encryption schemes. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *AFRICACRYPT 23*, volume 14064 of *LNCS*, pages 79–104. Springer, Cham, July 2023.

LMP17. Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing multi-key security degradation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 575–605. Springer, Cham, December 2017.

LP18. Atul Luykx and Bart Preneel. Optimal forgeries against polynomial-based MACs and GCM. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 445–467. Springer, Cham, April / May 2018.

MI15. D. McGrew and K. Igoe. AES-GCM Authenticated Encryption in the Secure Real-time Transport Protocol (SRTP). RFC 7714, December 2015.

Mou21. Nicky Mouha. Review of the advanced encryption standard. NIST Interagency/Internal Report 8319, National Institute of Standards and Technology, July 2021.

MV04. David A. McGrew and John Viega. The security and performance of the Galois/counter mode (GCM) of operation. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 343–355. Springer, Berlin, Heidelberg, December 2004.

Nan18. Mridul Nandi. Bernstein bound on WCS is tight - repairing luykx-preneel optimal forgeries. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 213–238. Springer, Cham, August 2018.

Nat25. National Institute of Standards and Technology. Pre-draft call for comments: GCM and GMAC block cipher modes of operation. NIST SP 800-38D Rev. 1 (Initial Preliminary Draft), January 2025.

Nat26. National Institute of Standards and Technology. Second pre-draft call for comments: GCM and GMAC block cipher modes of operation. NIST SP 800-38D Rev. 1 (2nd Preliminary Draft), June 2026.

NOMI15. Yuichi Niwa, Keisuke Ohashi, Kazuhiko Minematsu, and Tetsu Iwata. GCM security bounds reconsidered. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 385–407. Springer, Berlin, Heidelberg, March 2015.

NSS25. Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. The multi-user security of GCM-SST and further enhancements. In Sang Kil Cha and Jeongeun Park, editors, *ISC 2025*, volume 16186 of *LNCS*, pages 45–65. Springer, Cham, October 2025.

NSS26. Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Tight multi-user security of CCM and enhancement by tag-based key derivation. In Kangkook Jee and Aniket Kate, editors, *Applied Cryptography and Network Security - 24th International Conference, ACNS 2026, Stony Brook, NY, USA, June 22-25, 2026, Proceedings, Part I*, volume 16571 of *Lecture Notes in Computer Science*, pages 211–240. Springer, 2026.

Pat09. Jacques Patarin. The "coefficients H" technique (invited talk). In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, Berlin, Heidelberg, August 2009.

Res26. E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 9846, July 2026.

SCM08. J. Salowey, A. Choudhury, and D. McGrew. AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288, August 2008.

SI26. Rentaro Shiba and Tetsu Iwata. Multi-key security in the quantum world: Revisiting tweakable even-mansour and FX. *IACR Transactions on Symmetric Cryptology*, 2026(2), 2026.

SJS16. Vladimir Soukharev, David Jao, and Srinath Seshadri. Post-quantum security models for authenticated encryption. In Tsuyoshi Takagi, editor, *PQCrypto 2016*, pages 64–78. Springer, Cham, 2016.

TMC+25. Meltem Sönmez Turan, Kerry A. McKay, Donghoon Chang, Jinkeon Kang, and John Kelsey. Ascon-based lightweight cryptography standards for constrained devices: Authenticated encryption, hash, and extendable output functions. NIST Special Publication 800-232, National Institute of Standards and Technology, August 2025.

TT21. M. Thomson and S. Turner. Using TLS to Secure QUIC. RFC 9001, May 2021.

VM05. J. Viega and D. McGrew. The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP). RFC 4106, June 2005.